Doctoral Dissertation

Study on language processing methods for supporting understanding and using multiple legal documents

Tho Thi Ngoc LE

Supervisor: Associate Professor Nguyen Minh Le
(Co-supervise with Professor Akira Shimazu)

School of Information Science
Japan Advanced Institute of Science and Technology

September, 2015

Abstract

Law plays a significant role in governing our society and business. The system of legal documents in every country is often complicated with various kinds of documents, which are modified frequently to reflex the changing in situations of society/business, or to make the law more completed. Practically, the performance of retrieving legal information is still low when using the traditional strategy. Heretofore, the best solution to improve the performance is the exploiting a knowledge-base in retrieval. Nevertheless, the resource of knowledge-bases is not at hand and manually making knowledge-base is very expensive. For that reason, there is a requirement of automatic constructing of a knowledge-base to improve the performance of legal information retrieval. In addition, the contents and structures of legal documents are often complicated. Therefore, searching and reading legal documents is not easy for both normal citizens and legislators. We motivate to support the retrieving task by constructing the legal knowledge-base automatically; and, to help the readers by providing a hierarchical structure of legal indices which structurally yields the important information of legal documents. We divided the generation of the hierarchical structure into two main tasks: extracting legal indices and discovering relations among these indices.

The first task, extracting the indices which yield the main contents of legal documents, is treated as the problem of keyphrase extraction. We explored this extraction problem on two languages: Japanese and English. In the Japanese legal context, the legal indices are words, phrases and clauses. Since Japanese keyphrases are found in chunks and clauses, we approach index extraction using structural information of Japanese sentences, i.e. chunks and clauses. In English text, however, the chunk information does not really help improving the extraction performance because English chunks include words that cause noise in keyphrases. In the literature, current studies often extract English keyphrases by collecting adjacent important adjectives and nouns. Analysis on the data shows that keyphrases also contain other kinds of words. Hence, we proposed a solution to improve the extraction performance by involving new kinds of words to keyphrases.

The second task, constructing the relations among the indices, is treated as the problem of legal ontology construction. We proposed an approach to extract the super/subordinate relation between each pair of concepts individually based on directional similarity. The relations among a set of legal indices are represented in a directed graph and the hierarchical structure of indices is simply exported from this graph. We adopted

this proposal to the Japanese National Pension Act document. The resulted hierarchical structure is compared to an annotated legal ontology on the number of correct relations.

In this dissertation, there are two main contributions: novel approaches to extract keyphrases from Japanese and English text and novel approach to discover relationships among legal concepts in the construction of Japanese legal ontology. Our study serves as the necessary steps to construct the knowledge-based for legal information retrieval. In addition, the hierarchical structure of legal indices also serves as a structural summary of the main concepts, which enables the readers understand the relations among the legal concepts.

Keywords: legal engineering, unsupervised approach, keyphrase extraction, hierarchical index, ontology construction.

Acknowledgments

First of all, I wish to express my best sincere gratitude to my principal advisor, Professor Akira Shimazu of Japan Advanced Institute of Science and Technology (JAIST), for his constant encouragement, support and kind guidance during my PhD course. He has gently inspired me in researching as well as patiently taught me to be strong and self-confident in my study.

I would like to express the best thanks to my supervisor, Associate Professor Nguyen Le Minh of JAIST, for his fruitful discussions.

I would like to express the best appreciation to my sub-supervisor, Associate Professor Kiyoaki Shirai of JAIST, for helpful comments and excellent research environment.

I am grateful Professor Ho Tu Bao of JAIST for helpful discussions on my minor research.

I would like to thank Professor Satoshi Tojo for his comments on this dissertation.

I would like to express my sincere thanks to English teachers at Global Communication Education Department of JAIST, especially Professor W.R. Holden and J. Blake, for helping me improve writing skill.

I would like to dedicate this achievement to my respected parents, father Hoi and mother Sau, as well as my beloved two sisters, Ha Thu and Ngoc Mai. My family have consistently energized my spirit when studying abroad.

I would like to thank all of my friends, especially T.V.X. Phuong, for cheering me up whenever I have difficulty.

I also deeply acknowledge "Graduate Research Program" (GRP) and "JAIST Research Grant for Students" programs at JAIST that support me the finance during three years to research and present at the conference.

I would love to devote my sincere thanks and appreciation to all of my colleagues in Shirai's laboratory and Nguyen's laboratory.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In modern life, the law plays a vital role to keep all activities in our society and business operated smoothly. The law guarantees the peace, personal freedom and social justices by regulating the human behaviors in all aspects of the life, such as, economic, politic, social security, defense, education, culture, technology, environment. Without the law, our society would be chaos since the law provides a framework for the sustainment of our society. For business, the law establishes the standards, maintains the order, resolves disputes, and protects liberties/rights. McBride [McB10] describes four essential functions of the law as follows:

- Defending us from evil;
- Promoting the common good;
- Resolving disputes over limited resources;
- Encouraging people to do the right thing.

The contents of the laws are documented to popularize the law to everyone. Hence, the system of legal documents is very important in all countries and organizations in ruling the society. Corresponding to a lot of activities and resources, there are many kinds of legal documents:

- Public legal documents, published by the government or administrative organizations, which describe the contractual relationships or grant some rights, such as constitution, rules, codes, and ordinances.
- Minutes which record the legal activities such as reports of congress or courts.
- Civil documents which are issued or composed for civil purposes such as certificates of birth, marriages, contracts, and wills.

The scope of this dissertation covers only the public legal documents. Hence, the term *legal documents* in the context of this study means the documents which are published by the government or administrative organizations.

In general, the system of legal documents has the following characteristics:

- The system of legal documents includes a vast number of documents;
- Legal documents are frequently updated to keep up with the changes of our society;
- Each legal document is composed by long complicated sentences to describe how things work rely upon other parts in the same documents or in the other documents;
- The legal terms are made to have the unique meanings.

Due to the information load and the complexity of their contents, managing and reading legal documents are difficult for both legislators and normal citizens. In addition, the legal terms have unique meanings to the law and the sentences in legal documents are complex, hence the normal citizens may be confused when reading the information. Furthermore, the readers may be driven to different sections of the same document or to other documents when looking up the relevant information. Even the readers are professional, e.g. the legislators or the lawyers, they may be driven to a maze of information since the number of documents is too large. Additionally, the In such situations, many techniques have been applied to support the activities that relates to the legal documents.

Legal engineering [Kat07] is a research field which focuses on methodology to make, analyze and maintain legal documents as well as methodology to develop law-based information systems. Retrieving legal information from legal documents is basic in Legal Engineering. In this era, when most of information are documented digitally and stored in a database, the searching process is easier by applying the retrieving techniques.

Maxwell and Schafer [MS08] outline two broad approaches for legal information retrieval: natural language processing based approaches and knowledge engineering based approaches. Using natural language processing based approaches, the traditional way of information retrieval, there are many commercial search engines for retrieving the legal information such as WestLaw[1], LexisNexis[2] or FindLaw[3]. However, Blair and Maron [BM85] claim that traditional strategy, i.e. Boolean technique, works unsatisfactorily for retrieving legal information at 20% relevant documents while researchers believed that the

[1]WestLaw Search site can be found at `https://www.westlaw.com/` or `http://legalsolutions.thomsonreuters.com/`

[2]LexisNexis Search site can be found at `https://www.lexisnexis.com/search.aspx`

[3]FindLaw site can be found at `http://www.findlaw.com/`

performance should be 75% on general text. For knowledge engineering approaches, Saravanan et al. [SRR09] show that the retrieval performance is improved up to 89% using legal ontology and query enhancement. In practice, the knowledge-base is not available for all domains. Additionally, constructing legal ontologies to serve as knowledge-base for retrieving information requires the annotation of the concepts and its relations from the legal documents. On the other hand, the legal concepts are not easy to capture since they are not only single words but also compound words, phrases or clauses. Furthermore, the relations among the concepts are based on the semantics which reflects the real life. For those reasons, constructing a knowledge-base is a challenge.

In this context, both the specialists and non-specialists require a tool to support the activities related to legal documents. From the standpoint of the professionals, to support the process of legal information retrieval, there is a demand of constructing a knowledge-base. From the standpoint of the normal citizens, there is a need of a structural view that helps the readers to capture the main concepts in the legal documents as well as to understand the relations among those legal concepts. To respond to the requirements from the standpoints them, we introduce a hierarchical structure of legal indices which is a knowledge-base and can be considered as a hierarchical summary for the readers to understands the contractual relations among legal concepts.

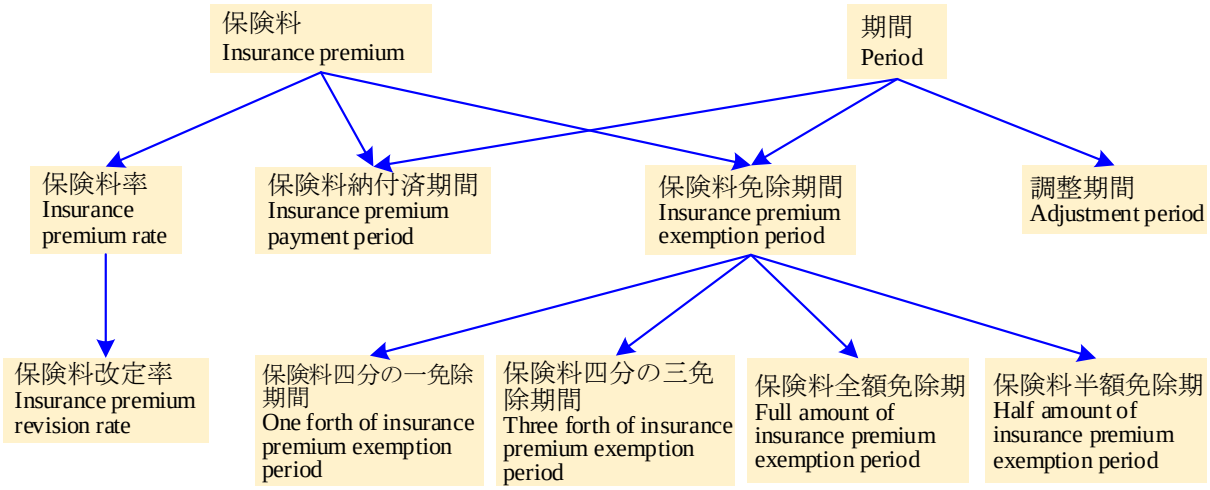


Figure 1.1: An example of hierarchical structure of legal indices in Japanese National Pension Act.

An example of the hierarchy which we target is shown in Figure 1.1. From this hierarchy, it is easier for the readers to figure out the semantic relations between every pair of concepts. Constructing the hierarchical structure of legal indices is a new task in Legal Engineering. We expect that the hierarchy could support the readers understanding the general contents of the legal documents. In addition, this structure could serve as a

knowledge-base to support the legal information retrieval task.

Problem Statement

Before clarifying the problems in consideration of this dissertation, we describe the structure of the hierarchy to have the following characteristics:

- The structure in consideration is a hierarchy that is similar to tree-view structure. In which the nodes in lower levels are subsumed by the nodes in higher levels on the meaning aspect. By this characteristic, the hierarchy of legal indices is recognized as a concept classifier which decomposes a general concept to many specific concepts.
- The difference to the tree-view structure is that, not only the parent nodes have many child nodes, but the child nodes can also have many parent nodes.
- Each node of the hierarchy is a keyword concept and the whole hierarchical structure is a summary of documents(s) in form of bag-of-words.

To construct the hierarchical structure of legal indices, we proceed two main tasks: extracting the legal concepts which yield the main contents of the legal documents; and discovering the relationships among the above legal concepts. The first task, extracting the legal concepts, is tackled as a problem of automatic keyphrase extraction. Two languages, Japanese and English, are made as objects for keyphrase extraction. The second task, discovering the relationships among legal concepts, is treated as the problem of legal ontology construction. For both tasks, due to the availability of the annotated data, we approach the solutions by unsupervised methods. Therefore, this dissertation is going to run into the following challenges:

- For keyphrase extraction from Japanese text, a new solution is required since most of unsupervised approaches are proposed for English text.
- For keyphrase extraction from English text, since current approaches involve only adjectives and nouns, our goal is to introduce new kinds of words to keyphrases and improve the extraction performance.
- For generating the hierarchical index, the relations among legal concepts are not easily brought to light using lexical matching, but the semantic relations among legal concepts should be extracted.

Contributions

This dissertation dedicates to the Legal Engineering and Natural Language Proceeding three contributions:

1. We proposed a novel method to extract keyphrases from Japanese legal text. In this research, legal indices are not limited to single-word keywords and compound-word (or phrase) keywords, they are also clause keywords. We approach index extraction using structural information of Japanese sentences. Based on the assumption that legal indices are composed of important tokens from the documents, extracting legal indices is treated as a problem of collecting chunks and clauses that contain as many important tokens as possible.

2. We made a motion on a novel unsupervised approach for English keyphrase extraction by involving new kinds of words to English keyphrases and improving the extraction performance. The current studies often extract keyphrases by collecting adjacent important adjectives and nouns. However, keyphrases actually include other kinds of words such as present/past participles, comparative/superlative adjectives and cardinal numbers. Therefore, we proposed to use the parsing information as a solution to improve the extraction performance by involving new kinds of words to keyphrases.

3. We introduced a proposal to discover the relations among the concepts for automatic legal ontology construction. This work serves as effort in extracting relationships among legal concepts for automatic legal ontology learning, in which super/sub-ordinate relations are considered. In this study, the super/sub-ordinate relations are discovered based on directional similarity.

Dissertation Outline

The work flow of our research is illustrated in Figure 1.2. This dissertation is detailed as following chapters:

Chapter 1 (this chapter) introduces the research context, clarifies the problems and summarizes the contributions of this study.

Chapter 2 presents the background about weighting schemes and semantic similarity which are employed in our proposed approaches.

Chapter 3 addresses the problem of automatically extracting legal indices which express the important contents of legal documents. Legal indices are not limited to single-word keywords and compound-word (or phrase) keywords, they are also clause keywords.

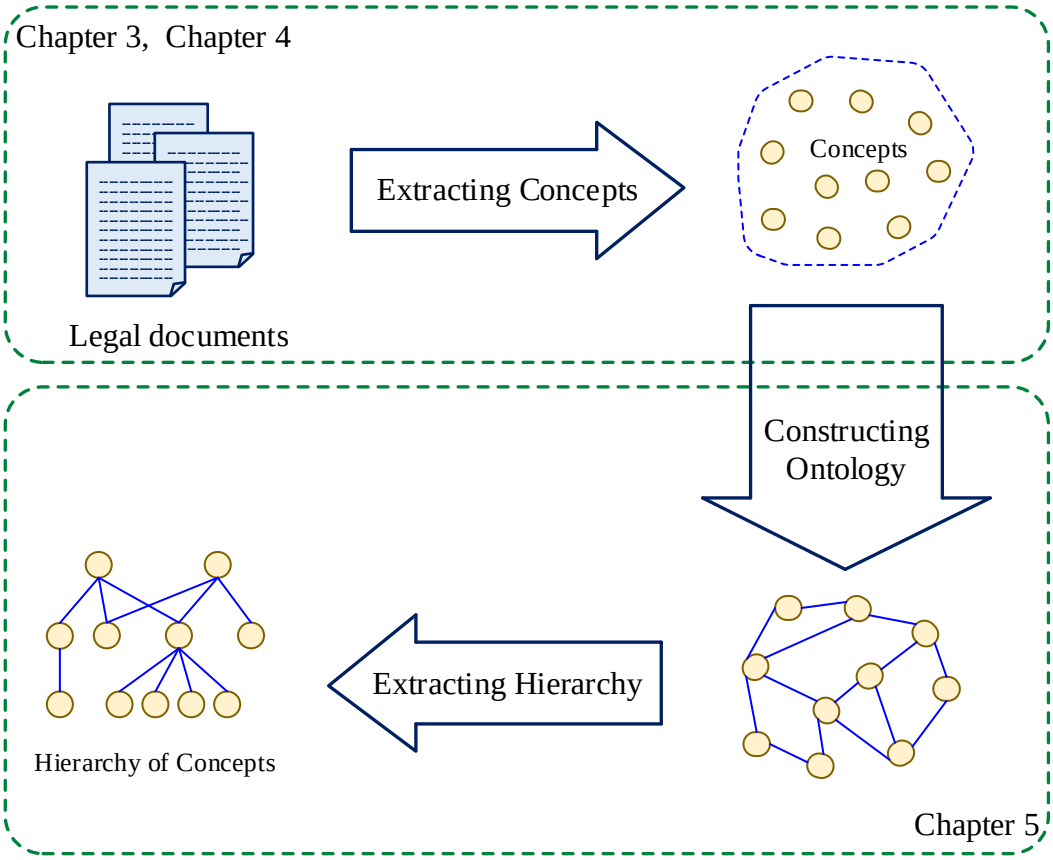


Figure 1.2: The work flow of our study in this dissertation.

We approach index extraction using structural information of Japanese sentences. Based on the assumption that legal indices are composed of important tokens from the documents, extracting legal indices is treated as a problem of collecting chunks and clauses that contain as many important tokens as possible. Each token is assigned a weight which are statistical scores to indicate its importance. The importance of a chunk or clause is determined based on the average weights of tokens included in that chunk or clause. Then, highly weighted chunks and clauses are recognized as the indices for legal documents.

Chapter 4 presents the solutions to extract keyphrases, in general, from English text. This chapter describes the adaption of weight averaging method to extract keyphrases in English text. Current studies often extract keyphrases by collecting adjacent important adjectives and nouns. However, keyphrases are actually contain other kinds of words such as present/past participles, comparative/superlative adjectives and cardinal numbers. Even so, incorporating such kinds of words to the noun phrase patterns is not a solution to improve the extraction performance. Therefore, we propose a solution to improve the extraction performance by involving new kinds of words to keyphrases. First, keyphrase candidates are extracted from noun phrases using syntactic information which

is obtained by shallow and deep parsing. Second, candidates are then associated with weights to indicate their importance in documents. The weight of a noun phrase candidate is computed as the average of the weights of tokens in it. Finally, the top weighted candidates in each document are selected as keyphrases for that document. We have experimented on four public corpora to demonstrate that our proposal improve the performance of keyphrase extraction and new kinds of words are introduced to keyphrases.

Chapter 5 presents our proposal to construct the hierarchical structure of legal indices. This work serves as effort in discovering relationships among legal concepts for automatic legal ontology learning, in which super/sub-ordinate relations are considered. With the indices extracted by the approach described in Chapter 3, we discover their relationships by language processing method. We propose an approach to extract the super/sub-ordinate relation between each pair of concepts individually based on directional similarity. The relations among a set of legal indices are represented in a directed graph and the hierarchical structure of indices is simply exported from this graph.

Chapter 6 summarizes the contributions and introduces the future directions from this dissertation.

Chapter 2

Background

Along the given problem statements and our solutions, we first introduce the measurements which will be applied in our approaches.

2.1 Weighting Schemes

We introduce two weighting schemes which are used in this dissertation: TF-IDF and Okapi BM25.

2.1.1 TF-IDF

The term *TF-IDF* stands for Term Frequency - Inverse Document Frequency. TF-IDF is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus [RU11]. This weighting scheme is widely used in information retrieval and data mining.

Term Frequency (TF) is a score indicating the importance of a term t in a document d. The simplest way to use term frequency of a term t is the raw frequency $f_{t,d}$ which is obtained by counting the number of its occurrences in the document d. In practice, there are many variations of term frequency:

- Boolean frequency: $\begin{cases} tf_{t,d} = 1 \text{ if } t \text{ occurs in } d\text{; or} \\ 0 \text{ otherwise.} \end{cases}$
- Logarithmic scaled frequency: $\begin{cases} tf_{t,d} = 1 + \log f_{t,d}\text{; or} \\ 0 \text{ if } f_{t,d} = 0. \end{cases}$
- Augmented frequency:

$$tf_{t,d} = \frac{0.5 + 0.5 \times f_{t,d}}{\max\{f_{w,d} : w \in d\}}$$

- Normalized frequency:

$$tf_{t,d} = \frac{f_{t,d}}{\max\{f_{w,d} : w \in d\}}$$

Inverse Document Frequency (IDF) score [Jon72] measures how important of a word t in a collection of documents D:

$$idf_{t,D} = \log \frac{|D|}{df_{t,D}}$$

where, $|D|$ is the total number of documents in the collections D, and $df_{t,D}$ is the number of documents containing term t. Due to the properties of logarithms, IDF scores of rare terms are high while IDF scores of of frequent terms are low.

TF-IDF score of a word is calculated as the product of its term frequency and inverse document frequency:

$$tfidf_{t,d,D} = tf_{t,d} \times idf_{t,D}$$

2.1.2 Okapi BM25

Okapi BM25 [RWJ+94] is a ranking function used by search engines to rank matching documents according to their relevance to a given search query. This weighting scheme is used in information retrieval. This weighting scheme measures the relevance of a query to a document. Given a query Q, containing keywords $q_1, q_2, \cdots, q_n$, Okapi BM25 score of a document d, which is drawn from a collection D, to the query Q is computed as:

$$bm25(d, Q) = \sum_{i=1}^{n} idf(q_i) \cdot \frac{f_{q_i,d} \cdot (k_1 + 1)}{f_{q_i,d} + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl})}$$

where, $f_{q_i,d}$ is the term frequency of the keyword q_i in the given document, $|d|$ is the length of the document d in words, and $avgdl$ is the average document length of documents in D. k_1 and b are parameters, usually chosen as $k_1 \in [1.2, 2.0]$ and $b = 0.75$. $idf(q_i)$ is the inverse document frequency of keyword q_i which is usually computed as:

$$idf(q_i) = \log \frac{N - df_{q_i} + 0.5}{df_{q_i} + 0.5}$$

where df_{q_i} is the number of documents in the collection D containing keyword q_i.

2.2 Text Similarity

Text similarity (or text relatedness) is a concept measuring the degree of overlapping in meaning between words, sentences, paragraphs, or documents in general. Measures of text semantic similarity have been used in many applications of NLP and related areas. One of earliest applications of text similarity is the vectorial model [SL68]. Text similarity has been used in many problems in NLP such as text classification, word sense disambiguation, extractive information, and text summarization. Specifically, given two texts (words, sentences, documents), the purpose of measuring text similarity is to figure out a score indicate their relations in meaning.

The simplest approach to find the similarity between two text segments is to use *lexical matching* method, and compute the similarity score based on the number of lexical units that occur in both input texts. To improve this simple method, weighting and factorizations [SB88] are considered, such as removing functional words (stop words), part of speech tagging, longest subsequence matching. However, the semantic of text is still hard to capture. For example, with two input *I have a dog* and *I own an animal*, lexical matching approach fails to discover the link between *dog* and *animal*, and unaware the identical meaning of *have* and *own* in this context. So, the purpose of finding text similarity score is not only take into account the similarity on the word surface but also on the semantic meaning.

To overcome the limit of semantic in lexical approaches, corpus-based and knowledge-based approaches use a large corpus and thesaurus to capture the semantic aspect of word [Tur01, LC98, WP94] based on the probability and statistics of words in input text. These semantic metrics have been successfully applied to NLP tasks such as word sense disambiguation [PBP03], and synonym identification [Tur01]. The vector-based approach is also a common choice to compare two strings of text in Information Retrieval systems [MBK00]. This approach represents a document as a vector, then comparing a pair of documents is equivalent to compute the distance or similarity of a pair of vector (e.g Euclidean, cosine...).

Another well-known method to compute similarity with corpus-based is the Latent Semantic Analysis (LSA) [LD97]. LSA is a high-dimensional linear association model, it analyses a large corpus of natural language text and generates a representation that get the similarity of words and text messages.

We distinguish text similarity to two main levels. The basic level is the similarity between words is mentions in Section 2.2.1. More advance in similarity is the semantic similarity between sentences or document is given in Section 2.2.2.

2.2.1 Similarity of Words

There is a large number of word-to-word similarity metrics that were proposed using distance-oriented measures computed from semantic networks, or using metrics based on models of distribute similarity learned from a large thesaurus.

The approach using distance-oriented measures computed the similarity of words from semantic networks [LC98] such as WordNet[1] [Mil95, Fel98]. This kind of metric considers the words as concepts and calculates the similarity of concepts based on the distance of them on the semantic networks. We recall some common metrics proposed in previous work based on WordNet, such as:

- Leacock & Chodorow similarity [LC98], the length of the shortest path between two concepts in WordNet is exploited using node counting and the maximum depth of taxonomy D.

$$Sim = -\log \frac{length}{2D}$$

- Lesk similarity [Les86], the similarity of two concepts is defined as a function overlap between the corresponding definitions in dictionary.

- Wu & Palmer similarity [WP94], the similarity of two concept is measured by the depth of two concepts in the taxonomy and the depth of the least common subsumer (LCS).

$$Sim = \frac{2 \times depth(LCS)}{depth(concept_1) + depth(concept_2)}$$

- Resnik similarity [Res95] combines the probability of encountering an instance of LCS to the information content (IC)

$$Sim = IC(LCS)$$

  in which information content IC of a concept c is defined as:

$$IC(c) = -logP(c)$$

  where $P(c)$ is the probability of occurrences of instances of concept c in a large corpus.

- Lin similarity [Lin98] add a normalization factor consisting of the information con-

[1] http://wordnet.princeton.edu/

tent of the input concepts.

$$Sim = \frac{2 \times IC(LCS)}{IC(concept_1) + IC(concept_2)}$$

- Jiang & Conrath similarity [JC97], the similarity of two concepts are computed by combining a lexical taxonomy structure with corpus statistical information.

$$Sim = \frac{1}{IC(concept_1) + IC(concept_2) - 2 \times IC(LCS)}$$

Recently, when the free encyclopaedia Wikipedia[2] become a popular dictionary, most concepts have been defined well by the the world community. Wikipedia become a promise thesaurus to look up the definitions of concepts. So some works are based on Wikipedia to find the similarity between concepts, such as uses snippets from Wikipedia to calculate the semantic similarity between words by using cosine similarity and TF-IDF [ZWZ09]. Another use machine learning techniques to explicitly represent the meaning of any text as a weighted vector of Wikipedia-based concepts, and calculate the similarity between words as the cosine between the corresponding vectors [GM07].

Beside knowledge-based methods as introduced above, corpus-based methods are also explored for usage in measure the similarity. Such as PMI-IR uses Pointwise Mutual Information (PMI) and Information Retrieval (IR) to measure the similarity of pairs of words [Tur01] using data collected by information retrieval. PMI-IR is an unsupervised measure for the evaluation of the semantic similarity of words. It is based on word co-occurrence using counts collected over very large corpora. With LSA, term cooccurrences in a corpus are captured by means of a dimensionality reduction operated by a singular value decomposition (SVD) on the term-by-document matrix T representing the corpus.

2.2.2 Similarity of Sentences and Documents

The combination of word similarity might not reveal how similar of two sentences or two documents, because word is just small unit in sentences or documents. Even word stores significant meaning in sentence and document, its meaning may vary depending on the context and usages. Then, the similarity (relatedness) between two sentences or two documents is still a challenge in NLP because of the meaning of text may vary in different context, or the complex pragmatic of sentences or document depends on the their usages.

From the first stage, the text similarity between sentences or documents can be easy figured out by vectorial representation, then various improvements proposed recently for such

[2]http://www.wikipedia.org/

techniques towards inventing more sophisticated weighting schemes for the text words, such as TF-IDF and its variations [Aiz03]. Though those techniques achieve certain results, the semantic aspect is still remained to be researched more. Usually, word-to-word similarity can be extended to more general text similarity [CM05]. Co-occurrence method in word-to-word similarity is extended to pattern matching method [CM05] which is often used in text mining, this technique relies on the assumption that documents are more similar if they contain more words in common. The words, in turn, are also considered in concepts aspect, or the semantic similarity of words rather than the lexical similarity.

A measure of relatedness between text segments must take into account both the lexical and the semantic relatedness between words. *Omiotis* [TVV10], a thesaurus-based similarity method exploits only a word thesaurus in order to devise implicit semantic links between words, which measure of semantic relatedness between texts which capitalizes on the word-to-word semantic relatedness measure (SR) and extends it to measure the relatedness between texts. Other approach employs the sentence syntax as *SyMSS* [OSdCI11] to measure the similarity for short texts. SyMSS captures and combines syntactic and semantic information to compute the semantic similarity of two sentences. Semantic information is obtained from a lexical database and through a deep parsing process that finds the phrases in each sentence.

Chapter 3

Japanese Legal Index Extraction

This chapter addresses the problem of automatically extracting legal indices which express the important contents of legal documents. Legal indices are not limited to single-word keywords and compound-word (or phrase) keywords, they are also clause keywords. We approach index extraction using structural information of Japanese sentences, i.e. chunks and clauses. Based on the assumption that legal indices are composed of important tokens from the documents, extracting legal indices is treated as a problem of collecting chunks and clauses that contain as many important tokens as possible. Each token is assigned a weight which are statistical scores, e.g. TF-IDF and Okapi BM25, to indicate its importance. The importance of a chunk or clause is determined based on the average weights of tokens included in that chunk or clause. Then, highly weighted chunks and clauses are recognized as the indices for legal documents. The experimental results on Japanese National Pension Act data show that our proposed method achieves better performance (8.6% higher on F1-score) than TextRank, the most popular unsupervised method in extracting single-word and compound-word keywords. In addition, this approach is also applicable to extract clause keywords with high performance.

3.1 Introduction

Law plays a significant role in governing our society and business. In Oxford Dictionary of Language matters[1], the role of the law is referred as "the system of rules which a particular country or community recognizes as regulating the actions of its members and which it may enforce by the imposition of penalties." In other words, the law guarantees the peace, personal freedom and social justice by regulating the human behaviors in all aspects of the life (e.g. economic, politic, social security, defense, education, cul-

[1]See online version at `http://www.oxforddictionaries.com/`

ture, technology, environment, etc.). Legal documents are documents which state some contractual relationship or grant some rights. They are documents officially published by the government organizations (such as constitution, rules, codes, ordinances, etc.) or privately edited documents for specific purposes (such as wills, contracts and agreements). Hereafter, we refer to legal documents as the documents published by the government organizations. Each legal document is only a part of the legal system for a country or an organization.

The system of legal documents in every country is often complicated with various kinds of documents which are modified frequently to reflex the changing in situations of social/business, or to make the law more completed. The legal documents are complicated by their nature since they are frequently characterized by long complex sentences containing many specialized terms and expressions unique to the field of law, words that have unique meaning in a legal content, archaic language and words borrowed from foreign languages. Sentences in legal documents are completed sentences and often make references to other sections within a document or to other legal documents. In addition, the language of legal documents is generally very formal, and structure is more important than readability. Therefore, reading legal documents is not easy, especially for non-specialists. Legal documents are difficult to read because not only the structure and sentences are written to avoid ambiguity, but also the vocabulary are unique to this field. Additionally, the references in legal text make the readers have to navigate many legal documents to look up the related information. In such situations, many techniques have been applied to support the activities that relate to legal documents.

Legal engineering [Kat07] is a research field which focuses on methodology to make, analyze and maintain legal documents as well as methodology to develop law-based information systems. Retrieving legal information from legal documents is basic in Legal Engineering and this study describes a method for making appropriate indices for it. [MS08] outline two broad approaches for legal information retrieval: natural language processing based approaches and knowledge engineering based approaches. For natural language processing approaches, [BM85] claim that traditional strategy works unsatisfactorily for retrieving legal information at 20% relevant documents while researchers believed that the performance should be 75% on general text. For knowledge engineering approaches, [SRR09] show that the retrieval performance is improved up to 89% using legal ontology and query enhancement. However, constructing legal ontologies to serve as knowledge-base for retrieving information requires the annotation of the concepts and its relations from the legal documents. On the other hand, the legal concepts are not easy to capture since they are not only sing-word keywords but also phrase and clause keywords.

In fact, keywords are words that yield the main ideas or important content of a sen-

tence or a document. Automatic keyword extraction significantly contributes to a variety of applications in Natural Language Processing (NLP) such as automatic summarization, text classification, information retrieval, question answering systems, and metadata/index creation. In legal context, keywords also play important roles such as creating indexes for legal cases [BA99] or supporting the prediction role for legal concepts [AB03]. Previous studies employed supervised approaches [Tur00, FPW+99, Hul03] and unsupervised approaches [MT04, LLZS09] to extract keywords. Because most previous approaches have focused on English documents, they do not perform well when applied to Japanese documents due to the differences in syntactic features. There is a limited number of approaches that are specifically designed to extract Japanese keywords, such as supervised methodology with simple lexical matching[SFS97, SFS98, OM97, Mat99] and unsupervised methodology using statistical information[NM02, NYM03, NM03, YN05] for general news domain. The disadvantage of supervised approaches is the requirement of annotated data for training process which is a costly and time-consuming task. In addition, annotation on the legal documents is much harder than on general texts. Therefore, unsupervised approaches for Japanese legal keyword extraction present greater opportunities for innovative methods of extraction to be employed.

Motivating from these gaps, in this study, we address the problem of extracting legal indices which reveal the main contains of legal documents and serves as the concepts for legal ontology construction. Specifically, we focus on methods of processing Japanese legal documents to provide readers and legal editors the summary of legal documents in terms of keywords or clauses that refer to the summary of the documents. In other words, this study aims to extract the important legal information, namely *legal indices*, which are single-word keywords, compound-word keywords and clause keywords.

Most previous studies take into account words and phrases for keywords when identifying indices for documents. In this work, we extend the objective of identifying the legal indices to three kinds of keywords: single-word keywords, phrase keywords and clause keywords. The hypothesis to extract legal indices is that indices are combined by meaningful tokens in the documents. However, when using current approaches to combine adjacent tokens to form keywords, the extraction performance is not good since Japanese keywords are not always have a fixed number of tokens. Specifically, the extracted keyword may include extra tokens if the keyword size is long and vice versa. Therefore, we approach a new method for Japanese keyword extraction.

We propose an unsupervised keyword extraction approach based on structural information of Japanese sentence, i.e. *chunks* (*bunsetsu* segments) and clauses (*setsu*). In the initial stage, all sentences are separated into chunks or clauses. Next, a weight of each chunk or clause is calculated based on its constituent tokens. Then, candidate

chunks and clauses, which are beneficial to keyword extraction, are selected based on the weights. Lastly, the candidates are processed to obtain the desired keywords which served as legal indices. Results obtained by the experiments on Japanese legal documents show that our approach to keyword extraction achieves better performance, i.e. 8.6% higher on F1-measure, than the most popular method, TextRank [MT04], in extracting single-word and phrase keywords. In addition, we also achieve high performance when applying our approach to extract clause keywords, i.e. 76.6%. Hence, we are able to conclude that the proposed approach can be effectively applied to Japanese legal documents.

The rest of this chapter is organized as follows: Section 3.2 reviews some representative approaches for keyword extraction, Section 3.3 overview the Japanese linguistic knowledge, Section 3.4 explains our proposed method, Section 3.5 describes the experiments we conducted, and Section 3.6 concludes the proposed approach and future work.

3.2 Related Work

Keyword extraction supports many applications in legal informatics, such as indexing legal cases [BA99] for case-based reasoning systems and legal concept extraction for predictive roles [MA03, GA11, AB03]. In this study, we focus on extracting keywords for Japanese legal documents to support summarizing and indexing.

Previous research has described various methods for extracting keywords from documents automatically. Those approaches are divided into two categories: supervised and unsupervised approaches. Supervised approaches consider extracting keywords as classification problems; the classifier is trained to decide whether a given word is a keyword or not. There are many approaches to train the classifier, such as combining heuristic rules and a generic algorithm to train the classifier [Tur99, Tur00], or alternately using Naive Bayes method for training process [FPW+99], and adding syntactic features like n-grams, part-of-speech tags and NP chunks [Hul03] to improve the performance. However, supervised approaches require annotated data for the training process, which is very costly and time-consuming.

To avoid the necessity of annotated data, most recent research in automatic keyword extraction has concentrated on unsupervised approaches. A typical unsupervised approach usually has two steps: the first is to extract as many candidate words as possible; the second is to apply the fixed combination of adjective(s) and noun(s) to combine candidates and obtain keywords. Previous approaches have explored many ways to collect candidate words which are potentially benefit to keyword extraction.

The most popular unsupervised approach is graph-based ranking algorithm first proposed in TextRank by [MT04]. To collect candidate words, the graph-based ranking

algorithm assumes that a word is important if it either connects to several other words or it has connections to important words. For a given document, TextRank constructs a graph of words, in which the vertexes are words with certain part-of-speech tags, and edges between any two vertexes are determined when two words in vertexes appear in a co-occurrence window (usually with size of 2 for English text). Each vertex has an initial weight; this weight will be changed based on a vertex's relations with other vertexes during the ranking process, which iterates until convergence. After that, a cut-threshold is applied to collect highly weighted vertexes, which are considered to be candidates for keywords. Following the idea of TextRank, there are many variations on graph-based ranking approaches, such as SingleRank, ExpandRank [WX08b], CollabRank [WX08a], Degree-based Ranking [LL08], Topical PageRank [LHZS10], and a context-sensitive Topical PageRank [ZJH+11]. Another notable unsupervised approach to extract candidate keywords applicable for English text is based on cluster exemplars [MI03, MI04, LLZS09].

All of these approaches are currently applied to extract English keywords. Although some of them have been identified as language independent approaches, they actually do not perform well when applied to extracting Japanese keywords. Compare to English keyword extraction, there is a limited number of studies published in English investigating supervised keyword extraction approaches for Japanese documents. For example, Suzuki et al. [SFS97, SFS98] described the use of term weighting combining with domain identification to train the classifiers and extract keywords from radio news; Ogawa and Matsuda [OM97] proposed an approach to segmenting Japanese text and extracting the overlapping segments to obtain keywords; Mathieu [Mat99] re-implemented the supervised learning method proposed by Turney [Tur99] to extract Japanese keywords using Japanese syntax. The disadvantage of these supervised approaches is the requirement of Japanese annotated data for training the classifiers. Therefore, researchers have also investigated the unsupervised approaches for Japanese keyword extraction. Nakagawa et al. [NM02, NYM03, NM03] proposed an unsupervised approach which explores the statistics between a compound noun and its component single-nouns to extract keywords from Japanese scientific abstracts. They assign a score to each compound noun which is calculated by multiplying the scores of single-nouns, in turn, the score of a single-noun is the production of frequencies of bi-grams containing that single-noun. Yoshida and Nakagawa [YN05] involved perplexity [MS99] and term frequency to score nouns and extract keywords from news text.

国民年金事業の / 事務の / 一部は、/ 政令の / 定める / ところにより、/ 法律によって / 組織された / 共済組合、/ 国家公務員共済組合連合会、/ 全国市町村職員共済組合連合会、/ 地方公務員共済組合連合会又は / 私立学校教職員共済法の / 規定により / 私立学校教職員共済制度を / 管掌する / ことと / された / 日本私立学校振興・共済事業団に / 行わせる / ことが / できる。
(For part of affairs of the national pension business, pursuant to the provisions of a Cabinet Order, the government may entrust Mutual Aid Association, Federation of National Public Service Personnel Mutual Aid Associations, National Federation of Mutual Aid Associations for Municipal Personnel, Pension Fund Association for Local Government Officials, which are organized by laws, or the Promotion and Mutual Aid Corporation for Private Schools of Japan, managed by the Private School Personnel Mutual Aid System pursuant to the provisions of the Private School Personnel Mutual Aid Association Act, to perform it.)

Figure 3.1: An example of chunks separated from a Japanese legal sentence of Article 3 paragraph 2 in Japan National Pension Act (2007).

3.3 Japanese Linguistic Knowledge

The writing system of Japanese consists of four alphabets: *Hiragana* (ひらがな), *Katakana* (カタカナ), *Kanji* (漢字) and *Romaji* (Roman characters). Japanese tokens are placed adjacently without spaces. In sentences, Japanese words usually occur as groups of words, namely chunks. A chunk in Japanese is a language unit, usually a block of tokens and can be referred to as a *phrasal unit* or *bunsetsu* segment. As a sentence constituent, a chunk is the smallest inseparable group of tokens in a sentence. In Japanese, a chunk usually includes one *independent word*[2] and optionally contains zero or more than one *auxiliary word*[3]. Figure 3.1 presents an example for a Japanese legal sentence which has been separated into chunks.

Japanese sentences, especially Japanese legal sentences, are usually written in compound sentences whose structures are very complicated and include many clauses. Biber et al. [BJL+99] define that a clause is a unit structured around a verb phrase which is accompanied by one or more elements denoting the participants involved in the actions, states, the attendant circumstances, the attitude, etc. Basically, a Japanese clause, so called *setsu*, contains one verb phrase. A Japanese compound sentence consists of a main clause (or independent clause) and one or more subordinate clauses (dependent clauses).

[2] *Independent words* (analogous to *free morphemes* in English) are words which have meaning and can stand alone in a sentence. In Japanese, independent words can be: nouns, pronouns, verbs, adjectives, adjectival nouns, adverbs, adnominal adjectives, conjunctions, or interjections.

[3] *Auxiliary words* are analogous to bound morphemes in English. They usually follow independent words to express the variation in meaning or to make clear the relations between and among independent words. In Japanese, auxiliary words can be either auxiliary verbs or particles.

1: 国民年金事業の事務の一部は、
(Part of affairs of the national pension business)
2: 政令の定めるところにより、
(pursuant to the provisions of a Cabinet Order)
3: 法律によって組織された
(which are organized by laws)
4: 共済組合、国家公務員共済組合連合会、全国市町村職員共済組合連合会、地方公務員共済組合連合会又は
(Mutual Aid Association, Federation of National Public Service Personnel Mutual Aid Associations, National Federation of Mutual Aid Associations for Municipal Personnel, Pension Fund Association for Local Government Officials, or)
5: 私立学校教職員共済法の規定により私立学校教職員共済制度を管掌することとされた日本私立学校振興・共済事業団に
(the Promotion and Mutual Aid Corporation for Private Schools of Japan, managed by the Private School Personnel Mutual Aid System pursuant to the provisions of the Private School Personnel Mutual Aid Association Act)
6: 行わせることができる。
(may entrust (organizations listed in 4 and 5) to perform it.)

Figure 3.2: An example of a Japanese legal compound sentence which contains six clauses. These clauses are approximately separated.

The main clauses in Japanese are usually at the end of the sentences. Figure 3.2 shows an example of the above Japanese sentence splitting into six clauses and the last clause "行わせることができる。(may entrust (organizations) to perform it.)" is the main clause.

Actually, the boundaries provided in Figure 3.2 are approximation since the main subject of the sentence "一部 (part)" is at the ending of the first clause. Strictly, the clauses of sentence should be separated as in Figure 3.3 where the main subject "一部 (part)" is placed at the beginning of the main clause. Note that, in Japanese sentences, subjects and objects are sometimes not expressed in clauses. However, as they are still understood by the readers based on the context, they are called zero pronouns.

The chunks in a sentence are connected together to form clauses. Except for the main verb which is placed at the end of the sentence, each of the other chunks has dependency to another chunk in the sentence. Figure 3.4 shows the dependencies of the chunks in the above example sentence. In which, the organizations listed in chunks 8, 9, 10, 11, 18 are co-ordinate.

3.4 Extracting Japanese Legal Indices

As discussed in Section 3.1, legal indices can be single-token indices or multi-tokens indices. In case of multi-tokens indices, they can be phrases or clauses. Hence, we divide

1: 政令の定めるところによる
(pursuant to the provisions of a Cabinet Order)
2: 法律による
(by law)
3: 組織された共済組合、国家公務員共済組合連合会、全国市町村職員共済組合連合会、地方公務員共済組合連合会
(organized Mutual Aid Association, Federation of National Public Service Personnel Mutual Aid Associations, National Federation of Mutual Aid Associations for Municipal Personnel, Pension Fund Association for Local Government Officials)
4: 規定による
(pursuant to the provisions)
5: 私立学校教職員共済制度を管掌することとされた日本私立学校振興・共済事業団
(the Promotion and Mutual Aid Corporation for Private Schools of Japan which administers the Private School Personnel Mutual Aid System)
6: 国民年金事業の事務の一部は、共済組合、国家公務員共済組合連合会、全国市町村職員共済組合連合会、地方公務員共済組合連合会又は日本私立学校振興・共済事業団に行わせることができる。(This is the main clause)
(For the part of affairs of the national pension business, the government may entrust the Mutual Aid Association, Federation of National Public Service Personnel Mutual Aid Associations, National Federation of Mutual Aid Associations for Municipal Personnel, Pension Fund Association for Local Government Officials or the Promotion and Mutual Aid Corporation for Private Schools of Japan to perform it.)

Figure 3.3: An example of Japanese clauses which are strictly separated.

legal indices into three types:

- *Word indices* (hereafter referred as *keytokens*) which are keywords containing only one token;
- *Phrase indices* (hereafter referred as *keyphrases*) which are keywords containing more than one token but not include the verbs;
- *Clause indices* (hereafter referred as *keyclauses*) which are clauses having important meaning in the legal documents.

For convenience, *indices* and *keywords* are used in a general context when referring to all kinds of keywords, *single keywords* is used interchangeably with *keytokens*, and *compound keywords* is used when referring to keyphrases and keyclauses.

As the legal indices are keywords that reveal the important information of legal text, extracting legal indices is the problem of keyword extraction in which keywords are meaningful words, phrases and clauses. By observation, since almost all Japanese keytokens and keyphrases from legal documents occur in chunks, extracting keytokens and keyphrases

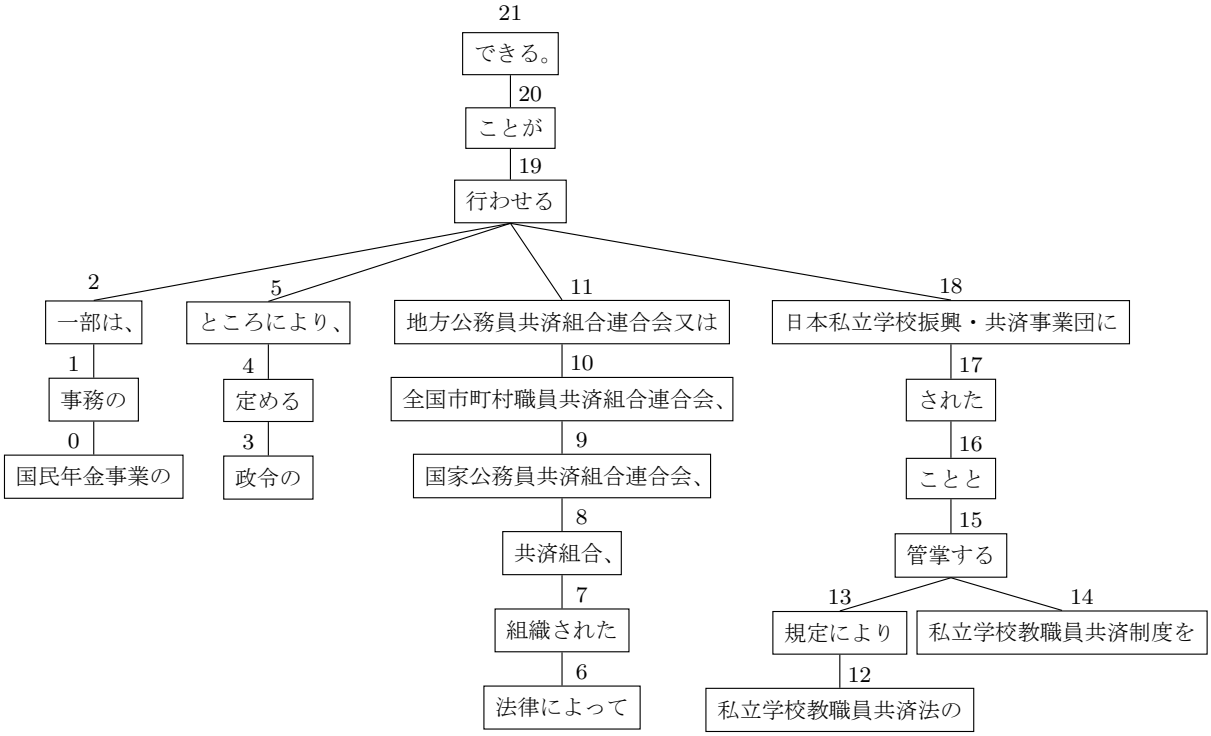


Figure 3.4: An example of Japanese dependencies connecting chunks in a sentence.

becomes a matter of finding chunks which determine the content of the document. In addition, since keyclauses are combined from important tokens, the extracting of keyclauses becomes the problems of finding clauses which contain as many important tokens as possible.

We propose an unsupervised approach that extracts the important chunks and clauses from documents. First, all sentences in a document are parsed into chunks and clauses. Second, all tokens in each chunk or clause are assigned weights to indicate their relevance in the document, then the average weight of those tokens expresses the significance of the chunk or the clause containing them. Third, candidates for indices are identified by collecting chunks and clauses which are recognized as important in the given document. Finally, those candidate chunks and clauses are post-processed to obtain keywords. The outline of the proposed approach is shown in Algorithm 1.

Given a Japanese legal document d, the desired output is a set of extracted indices. The task of Japanese clause boundary identification has been investigated with CBAP program[4] [MKKT04] and the task of parsing a Japanese text into smaller language units is assumed to be achieved by the existing Japanese Dependency Structure Analyzer tool [KM02]. Thus, we will not explain the processes of dividing a document into sentences,

[4]Note that, CBAP program identifies the approximate boundaries of clauses. Hence, in experiments, we do not use CBAP but separate clauses manually.

Algorithm 1: Legal Index Extraction Algorithm

input : A document d
output: Set of keywords K

S: all sentences in the document d;
C: all chunks and clauses in the document d;
Can: candidates for keywords in the document d;
$S \leftarrow$ Sentences in d
foreach *sentence* $s \in S$ **do**
    $C_s \leftarrow$ Parse sentence s for chunks and clauses;
    $C \leftarrow C \cup C_s$;
end
foreach *chunk (or clause)* $c \in C$ **do**
    $weight_c = CalculateWeight(c)$;
end
Determine threshold θ for candidates based on the input document d;
$Can \leftarrow$ Collect all chunk (and clause) $c \in C$ that satisfies $weight_c > \theta$;
$K \leftarrow$ Post-process(Can);

separating a sentence into clauses or chunks, and splitting a clause/chunk into tokens. The next subsections will specify in detail how the weights of tokens, chunks and clauses are calculated, how the thresholds for index candidates based on a given document are defined, and how Japanese legal indices are generated from candidate clauses and chunks.

3.4.1 Weights

The importance of a token is expressed by a score considering the context of a document and a collection of documents. In this study, we consider two weighting schemes: TF-IDF and Okapi BM25.

The Term Frequency - Inverse Document Frequency (TF-IDF) score is employed to express the significance of tokens under consideration to both a document and an overall document set. The Term Frequency (TF) score of a token expresses the importance of token within a single document. The TF score of a token t in a document d (denoted by $tf_{t,d}$) can simply be the number of occurrences of the token in a given document (denoted by f_t), or be calculated as the logarithmic scaled frequency:

$$tf_{t,d} = \log(f_t + 1)$$

The Inverse Document Frequency (IDF) score of a token indicates the importance of the token in a collection of documents D. The IDF score of a token is high if it is a rare token

in the collection of documents, but low if it occurs frequently. The IDF score of a token (denoted by $idf_{t,D}$) is calculated by:

$$idf_{t,D} = \log \frac{N}{df_t}$$

where df_t is document frequency, calculated by counting the number of documents that contain the token t, and N is the number of documents in D. The TF-IDF score (denoted by $tfidf_{t,d}$) of a token is defined as the product of its TF and IDF scores:

$$tfidf_{t,d} = tf_{t,d} \times idf_{t,D}$$

Okapi BM25 [RWJ+94] is ranking function used by search engine of information retrieval. For a token t, its Okapi score regarding to a document d in a collection of documents D is computed as follows:

$$okapi(t,d) = idf_{t,D} \times \frac{tf_{t,d} \times (k_1 + 1)}{tf_{f,d} + k_1 \times (1 - b + b \times \frac{|d|}{avgdl(D)})}$$

where $|d|$ is the length of document d in words, $avgdl(D)$ is the average document length of all documents in the collection D. k_1 and b are free parameters, usually chosen as $k_1 \in [1.2, 2.0]$ and $b = 0.75$. The inverse document frequency for Okapi score is usually computed as:

$$idf_{t,d} = \log \frac{N - tf_{t,d} + 0.5}{idf_{t,D} + 0.5}$$

The weight of chunks and clauses are calculated similarly. The weight of each chunk or clause is based on the weights of tokens that belong to that chunk/clause. The weight of a chunk/clause c is the average weight of all included tokens:

$$weight_c = \frac{\sum_{t \in T} weight_t}{|T|}$$

where $weight_c$ is the weight of chunk/clauses c, $weight_t$ is the weight of a token t ($t \in T$), and T is the set of tokens that belong to the chunk/clause. Note that the weights of stop-words affect the weights of the chunks. Hence, the set T may include some tokens which are beneficial to the weighting, such as a set of tokens whose weights are greater than zero, or a set of tokens which are not stop-words.

Table 3.1 illustrates the calculation of the weight of the chunk "私立学校教職員共済法の (of Private School Personnel Mutual Aid Association Act)," in which the weight of stopword "の (is)" is reset to zero and ignored when computing the average.

Table 3.1: Calculation for the weight of the chunk "私立学校教職員共済法の (of Private School Personnel Mutual Aid Association Act)."

Token	私立	学校	教職員	共済	法	の	**Avg.**
Weight	1.7	1.09	1.7	2.32	0.47	0	**1.45**

Similarly, the weight of the clause ""私立学校教職員共済制度を管掌することとするれる日本私立学校振興・共済事業団 (the Promotion and Mutual Aid Corporation for Private Schools of Japan which is managed by a Private School Personnel Mutual Aid System)" is also calculated based on the weights of its tokens as in Table 3.2. In this example, the weight of stopwords such as "を (at)" and "の (of)" are reset to zero since they do not contribute to the meaning of the legal indices. These stopwords are also not counted when calculate the average weight of the clause.

Table 3.2: Calculation for the weight of the clause "私立学校教職員共済制度を管掌することとするれる日本私立学校振興・共済事業団 (the Promotion and Mutual Aid Corporation for Private Schools of Japan which is managed by a Private School Personnel Mutual Aid System)."

Token	私立	学校	教職員	共済	制度	を	管掌	する	こと	と	**Avg.**
Weight	1.7	1.09	1.7	2.32	0.8	0	1.81	0.39	0.53	0	
Token	する	れる	日本	私立	学校	振興	・	共済	事業	団	**1.12**
Weight	0.39	0.44	0.55	1.7	1.09	0.88	0	2.32	0.87	0.43	

3.4.2 Thresholds

We define two types of thresholds: threshold θ_s for single keywords and threshold θ_c for compound keywords. Usually, the important tokens that appear in keywords have higher weights compared to the average weight of all tokens within a document. Candidate chunks/clauses are those which own as many potential tokens as possible. In addition, because the weight of a chunk/clause is the average weight of its tokens, if we view a chunk/clause as a large token, this token is a candidate when its weight is higher than the average of all tokens in the given document. We define a threshold θ_c for compound keywords based on the average weight of all tokens in the given document:

$$\theta_c = \beta \times \frac{\sum_{t \in T_d} weight_t}{|T_d|}$$

where $weight_t$ is the weight of a token t, a set T_d is all tokens appearing in the given document d, and $|T_d|$ is the number of elements in the set T_d. We added a coefficient

β to control the threshold in case we need more, or fewer, keywords. Note that the set $|T_d|$ includes all tokens which appear in the given document, including auxiliary and zero-weight tokens.

In the case of single keywords, a token is a keyword in a given document if it is recognized as very important in the context of that document. Thus, a token which is a candidate keyword should have a significantly higher score compared to other tokens that hold meaning in the document (i.e. hold a greater-than-zero weights). For this reason, a threshold θ_s for single keywords is defined based on the average weight of tokens whose weights are greater than zero:

$$\theta_s = \alpha \times \frac{\sum_{t \in {T^+}_d} weight_t}{|{T^+}_d|}$$

in which $weight_t$ is the weight of a token t, a set ${T^+}_d$ includes tokens whose weights are greater than zero in d, and $|{T^+}_d|$ is the number of elements in the set ${T^+}_d$. Similar to coefficient β, coefficient α is introduced to control the number of single keywords that will be extracted.

3.4.3 Post-Processing

This process is performed to remove unnecessary words at the beginning and ending of candidate clauses and chunks. Intuitively, a keyword must have meaning - it must be an independent word/clause.

In cases of keytokens and keyphrases, they should not include auxiliary words serving a grammatical role. Thus, we retain the independent words and remove all of the auxiliary words from the beginning and ending of chunk candidates. Table 3.3 lists the part-of-speech tags which are considered as independent words. The other tokens are treated as auxiliary words or irrelevant words. For example, particle "の (of)" is removed from candidate "私立学校教職員共済法の (of Private School Personnel Mutual Aid Association Act)" since this particle is considered as an auxiliary word.

In cases of keyclauses, we remove the tokens or phrases that serve specific grammatical roles. The post-processing process for extracting keyclauses is as follows:

(i) First, the ending tokens whose POS tags are 助詞 (particle) and 判定詞 (copula) in the clauses are removed, e.g. particle は (wa) which serves as topic marker in a sentence and copula だ (da) which serves as special word that combines the subject of a sentence and its description.

(ii) Second, the successors listed in Table 3.4 are removed from verbs and adjectives of clause candidates.

(iii) Third, the successors and nouns listed in Table 3.5 and Table 3.6 are removed if they are at the end of clause candidates.

(iv) Finally, the candidates which contain only one token after the above three steps are also eliminated since they are not clauses.

For example, the main clause of above example in Figure 3 is processed by removing successor "ことができる (be able to / may)" from the verb "行わせる (entrust ... to perform)."

Table 3.3: POS tags of independent words.

ChaSen POS Tag	**Description**	**Example**
名詞 -一般	Noun(general)	耳 (ear)
名詞 -固有名詞 -一般	Noun(proper.general)	光が丘 (Hikarigaoka)
名詞 -固有名詞 -人名 -一般	Noun(proper.name.general)	お市の方 (Oichinokata)
名詞 -固有名詞 -人名 -姓	Noun(proper.name.surname)	山田 (Yamada)
名詞 -固有名詞 -人名 -名	Noun(proper.name.firstname)	紀子 (Noriko)
名詞 -固有名詞 -組織	Noun(proper.organization)	NHK
名詞 -固有名詞 -地域 -一般	Noun(proper.place.general)	京都 (Kyoto)
名詞 -固有名詞 -地域 -国	Noun(proper.place.country)	日本 (Japan)
名詞 -非自立 -一般	Noun(bound.general)	こと (thing)
名詞 -サ変接続	Noun(verbal)	見学する (visit)
名詞 -形容動詞語幹	Noun(adjective -na)	安全 (safe)
名詞 -接尾 -一般	Noun(suffix.general)	印 (mark)
名詞 -接尾 -地域	Noun(suffix.place)	駅 (station)
名詞 -接尾 -サ変接続	Noun(suffix.verbal)	話 (story)
名詞 -接尾 -形容動詞語幹	Noun(suffix.adjective-na)	-的 (-tive)
形容詞 -自立	adjective–i(free)	苦い (near)
形容詞 -非自立	adjective -i(bound)	難い (difficult)
形容詞 -接尾	adjective -i(suffix)	-っぽい (like)
接頭詞 -名詞接続	prefix(+noun)	両 (both)

3.4.4 Extracting Keywords

Initially, candidates for keywords are chunks and clauses that have weights greater than the threshold for compound keywords θ_c. These candidates will be post-processed to obtain only meaningful independent words/clauses. Sometimes, only one token is left in a candidate after post processing. For keytoken and keyphrase extraction, each of the remaining tokens is then examined to determine whether it is a keyword by comparing its

Table 3.4: Successors to be removed from verbs and adjectives in clause candidates.

Successor	**Translation**	**Successor**	**Translation**
ことができるない	it is not possible (that)	べきものである	(that) should be
ことができる	it is possible (that)	べき	should
こととなるとき	when it becomes (that)	ものだとき	when (that)
つつ	while	ものとする	do (that)
ところによる	based on (that)	ものとみなす	regarding (that)
べきであった	should have been		

Table 3.5: Successors to be removed when occurring at the end of clause candidates.

Successor	**Translation**	**Successor**	**Translation**
につく (bare form) について (in text)	regarding to	におく (bare form) において (in text)	at (place or time)

Table 3.6: Nouns which serves as grammatical roles.

Noun	**Translation**	**Noun**	**Translation**
場合	(in) case	際	at (time), (in) case
とき	when	ため	for (purpose)
こと	that, which, thing	当時	at that time
事由	situation		

weight to the pre-determined threshold for single keyword θ_s. For keyclause extraction, the candidates which contain only one token are removed since they are not clauses.

3.5 Experiments

We used the Japanese National Pension Act (JNPA) as the corpus for our experiment. We first removed all headings and titles, so the data included only the sentences in legal articles of JNPA, and each sentence was placed in a separate line. JNPA contains 877 sentences where 99 keytokens, 109 keyphrases, 2,019 keyclauses are manually annotated by professionals[5]

[5]Persons who have experiences to annotate several legal documents including one for Japanese National Pension Act. They have extracted indices from the main part of JNPA.

3.5.1 Implementation

The Japanese parser tool Cabocha[6] [KM02] is utilized to decompose Japanese sentences into chunks and tokens. Though the Japanese clause boundary identification problem has been addressed in CBAP [MKKT04], its performance is limited in approximated boundaries of Japanese clauses. Therefore, in the context of long and complicated sentences of legal sentences, we cannot obtain satisfied performance. For that reason, the clause boundaries of legal sentences are annotated manually.

We employed two weighting schemes TF-IDF and Okapi BM25 as described in Section 3.4.1. Because IDF score can be different based for a reference corpus, two corpora were employed to calculate IDF scores of tokens. The first corpus was a representative of the legal domain and included 7,984 law documents[7]. The other corpus was a representative of the general domain and contained 496,997 news articles from Mainichi Shimbun newspapers published between 1991 and 1995. In addition, IDF score is also sensitive to the elements in the corpus on which it is calculated. Hence, the experiment was run on two original sets of corpora and three combinations of the same corpora to calculate the IDF scores of tokens in order to examine the effectiveness of IDF score on extraction performance:

(i) 7,984 Japanese legal documents;

(ii) 496,997 Mainichi Shimbun articles from years 1991 to 1995;

(iii) 7,984 legal documents and 496,997 news articles;

(iv) 7,984 legal documents and 111,497 news articles from the year 1995;

(v) 7,984 legal documents and 7,984 news articles from the year 1995.

The TF scores of tokens are counted on the whole JNPA document, i.e. including supplementary provisions. The two coefficients, α and β, controlling the number of keywords are set to 1. Table 3.7 and Table 3.8 show the correct indices extracted from the example sentence showing in Figure 3.1.

3.5.2 Evaluation

Since there is no previous work on extracting keyclauses, we separate the evaluation into two parts. The first part evaluates the performance of our proposed approach on

[6]Cabocha is available at `https://code.google.com/p/cabocha/`

[7]The law documents are obtained from the Japanese government web page corpus which is updated on July 1st, 2013.

Table 3.7: Example of Japanese legal keytokens and keyphrases.

Keyword	**Translation**
日本私立学校振興・共済事業団	Promotion and Mutual Aid Corporation for Private Schools of Japan
国家公務員共済組合連合会	Federation of National Public Service Personnel Mutual Aid Associations
地方公務員共済組合連合会	Pension Fund Association for Local Government Officials
私立学校教職員共済法	Private School Personnel Mutual Aid Association Act
国民年金事業	national pension business
共済組合	Mutual Aid Association
事務	affair
政令	Cabinet Order
規定	the provisions

Table 3.8: Example of Japanese legal keyclauses.

Keyclause	**Translation**
組織された共済組合、国家公務員共済組合連合会、全国市町村職員共済組合連合会、地方公務員共済組合連合会	organized Mutual Aid Association, Federation of National Public Service Personnel Mutual Aid Associations, National Federation of Mutual Aid Associations for Municipal Personnel, Pension Fund Association for Local Government Officials
私立学校教職員共済制度を管掌することとされた日本私立学校振興・共済事業団	the Promotion and Mutual Aid Corporation for Private Schools of Japan administers the Private School Personnel Mutual Aid System
国民年金事業の事務の一部は、共済組合、国家公務員共済組合連合会、全国市町村職員共済組合連合会、地方公務員共済組合連合会又は日本私立学校振興・共済事業団に行わせる	For the part of affairs of the national pension business, the government entrust the Mutual Aid Association, Federation of National Public Service Personnel Mutual Aid Associations, National Federation of Mutual Aid Associations for Municipal Personnel, Pension Fund Association for Local Government Officials or the Promotion and Mutual Aid Corporation for Private Schools of Japan to perform it.

extracting keytokens and keyphrases in comparison to a popular baseline. The second part shows the performance of our approach on extracting keyclauses from legal documents. This work follows the evaluation criteria of previous studies in automatic keyword extraction: *Precision*, *Recall* and *F1-score*. Precision score measures how many extracted keywords are correct and be calculated by the fraction of correct keywords among ex-

tracted keywords:

$$Precision = \frac{\#\text{ Correct keywords}}{\#\text{ Extracted keywords}}$$

Recall score measures how many correct keywords are extracted among annotated keywords and be computed by the fraction of correct extracted keyword comparing to manually assigned keywords:

$$Recall = \frac{\#\text{ Correct keywords}}{\#\text{ Annotated keywords}}$$

F1-score measures the test accuracy by considering the precision and recall scores. F1-score can be the harmonic mean of precision and recall:

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

3.5.3 Baseline Implementation

To evaluate the proposed method in extracting keytokens and keyphrases, we re-implemented the popular graph-based ranking algorithm TextRank [MT04] for Japanese keyword extraction. Although clustering-based approach [LLZS09] reports higher results compared to TextRank when extracting English keywords, it requires an extra resource (Wikipedia) to compute the similarities. Thus, we choose the TextRank algorithm because it does not require any external resource and is still widely used in automatic keyword extraction [WZO10, LHZS10, ZJH+11]. The overview of TextRank is given in Section 3.2.

In this evaluation, we tailored TextRank for Japanese keyword extraction. When parsing Japanese text, tokens with POS-tags described in Table 3.3 are recognized as nouns and adjectives. These tokens are added to the graph of tokens. The relation between any two tokens is established if they occur in a co-occurrence window. After ranking the vertices in the graph, a cut-off threshold T is used to get high ranked tokens from the graph as candidates for keywords. The cut-off threshold T indicates the number of candidate tokens from the graph of words. Then, keywords and keyphrases are extracted by combining graph vertices which adjacently appear in the document.

We recognize that the original parameter settings of TextRank does not work well for Japanese legal text. The size of co-occurrence window W determine the number of tokens in keywords and several Japanese keywords has more than two tokens. Therefore, we adjust the window size W taking values from $\{2, 3, 4, 5, 6, 7, 8, 9, 10\}$. On the other hand, Japanese legal keywords also include those which have only one token. To find the best performance of TextRank on Japanese legal text, we collect a percentage S of

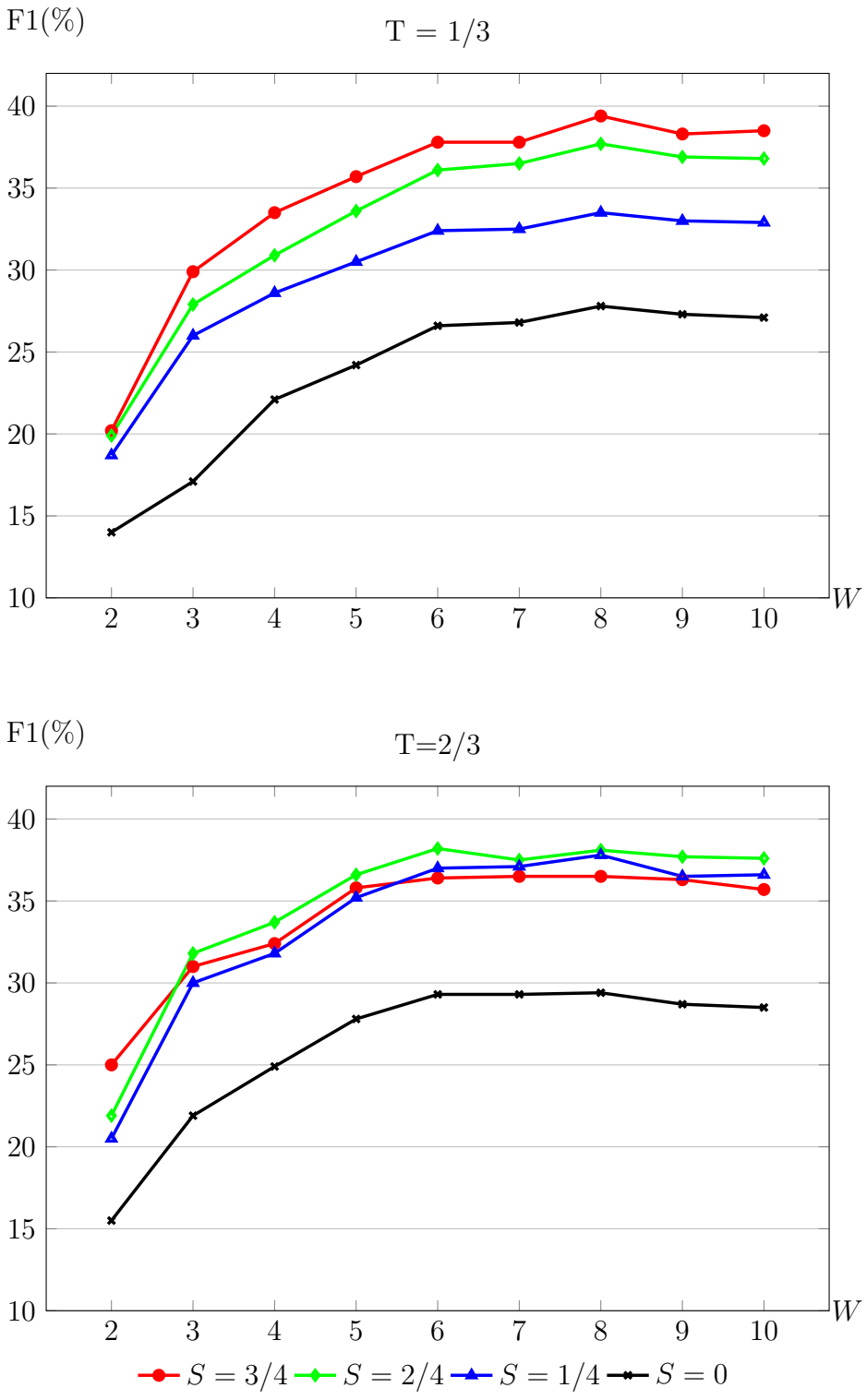


Figure 3.5: The performance of TextRank on JPNA data with different values of parameters.

highest ranked vertices for single-token keywords $S = \{0, 1/4, 2/4, 3/4\}$. In addition, since the cut-off threshold affects the percentage of candidates, we also investigate the cut-off threshold $T = \{1/3, 2/3\}$. Note that, both T and S are the percentages to collect highly ranked vertices from the graph of tokens. The difference is that T is the percentage to collect candidates for keyphrases while S is the percentage to collect candidates for keytokens.

The performances of TextRank with different parameters are sketched in Figure 3.5. With our re-specification of parameters, the original settings of TextRank are $W = 2$, $T = 1/3$ and $S = 0$. As demonstrated in Figure 3.5, TextRank achieves poor performance on JNPA data, i.e. $F1 = 14\%$, with its original setting. To make the evaluation more valid, therefore, we tune the parameters for best performance of TextRank. On JNPA data, when the highly ranked tokens in the graph are not considered as potential for keywords, the results are much lower than those considering such tokens. With cut-off threshold $T = 2/3$, performances of TextRank are approximately the same when $S \neq 0$. In all cases of parameter settings, we realize that TextRank reaches its stable performance when the co-occurrence window size is greater than or equal to 6. In our experiment on JNPA, TextRank achieves the best performance $F1 = 39.38\%$ with setting for parameters is $T = 1/3$, $S = 3/4$ and $W = 8$. We pick up some of the highest $F1$ scores as performances of TextRank for comparing with our proposed approach in extracting keytokens and keyphrases.

3.5.4 Performance Evaluation on Extracting Keytokens and Keyphrases

TF-IDF and Okapi BM25 are applied as weighting scheme to assign the importance to tokens. Since the parameter $k1$ has values in $[1.2, 2.0]$, we plot the performance of chunk-based approach with various options of values for $k1$ on different corpora in Figure 3.6. As shown in this figure, the higher of value $k1$, the better of performance. Therefore, we choose $k1 = 2.0$ when computing Okapi scores of the tokens.

Table 3.9 shows the performance of our proposed approach in comparison with TextRank. The highest scores at each evaluation criteria are shown in bold. Based on the F1-score presented in Table 3.9, most of F1-scores of the proposed approach are higher than TextRank. Hence, we conclude that the proposed chunk-based approach is better than the graph-based approach analyzed in this experiment. When comparing the highest F1-score, i.e. 47.97%, the proposed approach outperforms TextRank on F1-score, yielding a 8.6% improvement on overall evaluation.

The extraction performance depends on how weights are assigned to tokens. First, the

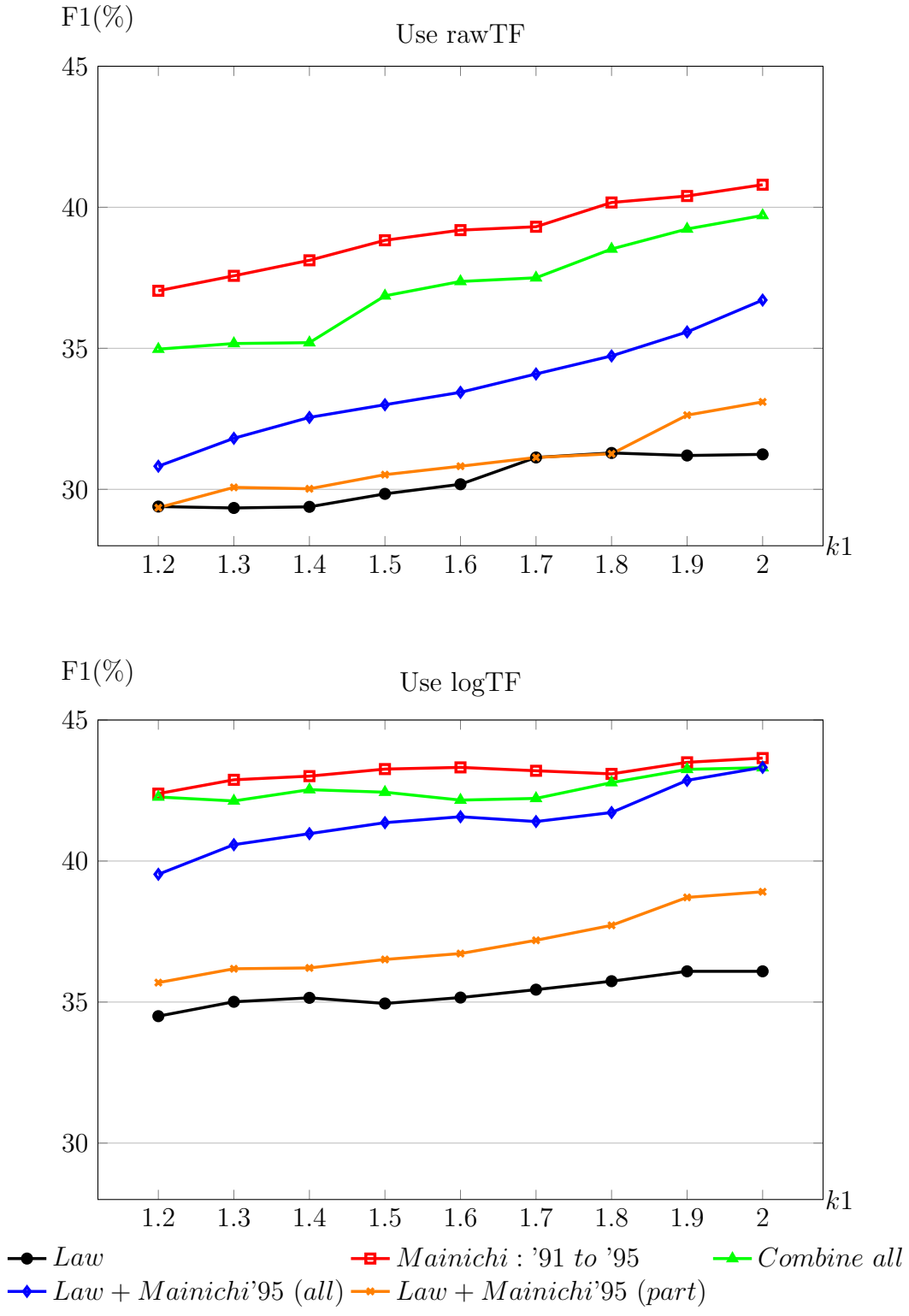


Figure 3.6: The performance of proposed approach on extracting keytokens and keyphrases from JPNA using Okapi BM25 weighting scheme.

Table 3.9: Results of chunk-based keyword extraction approach in comparison with graph-based ranking approach TextRank on JNPA data.

Methods		**#Extr.**	**#Corr.**	**Prec.**	**Rec.**	**F1**
TextRank						
$T = 1/3$, $S = 0$, $W = 2$		278	34	12.23	16.35	13.99
$T = 1/3$, $S = 3/4$, $W = 8$		432	126	29.17	60.58	**39.38**
$T = 2/3$, $S = 3/4$, $W = 6$		698	165	23.64	79.33	36.42
$T = 2/3$, $S = 3/4$, $W = 7$		697	165	23.67	79.33	36.46
Proposed approach using TF-IDF weighting scheme						
Corpus for IDF	TF					
Law	logTF	384	142	**36.98**	68.27	**47.97**
Law	rawTF	474	146	30.80	70.19	42.82
Mai:'91 to '95	logTF	562	174	30.96	83.65	45.19
Mai:'91 to '95	rawTF	643	165	25.66	79.33	38.78
Combine all	logTF	572	176	30.77	84.62	45.13
Combine all	rawTF	567	166	29.28	79.81	42.84
Law&Mai'95(all)	logTF	565	178	31.50	**85.58**	46.05
Law&Mai'95(all)	rawTF	616	162	26.30	77.88	39.32
Law&Mai'95(part)	logTF	464	161	34.70	77.40	47.92
Law&Mai'95(part)	rawTF	617	163	26.42	78.37	39.52
Proposed approach using Okapi BM25 weighting scheme: $k1 = 2.0$ and $b = 0.75$						
Corpus for IDF	TF					
Law	logTF	529	133	25.14	63.94	36.09
Law	rawTF	477	107	22.43	51.44	31.24
Mai:'91 to '95	logTF	548	165	30.11	79.33	**43.65**
Mai:'91 to '95	rawTF	493	143	29.01	68.75	40.80
Combine all	logTF	554	165	29.78	**79.33**	43.31
Combine all	rawTF	472	135	28.60	64.90	39.71
Law&Mai'95(all)	logTF	526	159	**30.23**	76.44	43.32
Law&Mai'95(all)	rawTF	424	116	27.36	55.77	36.71
Law&Mai'95(part)	logTF	414	121	29.23	58.17	38.91
Law&Mai'95(part)	rawTF	366	95	25.96	45.67	33.10

weighting schemes affect the performance of our proposed approach. As shown in Table 3.9, F1-scores of TF-IDF are likely higher than Okapi BM25 when using the same corpus and the same type of term frequency. Comparing the best F1-scores for each weighting scheme, TF-IDF is 4.3% higher than Okapi BM25. Second, there is a remarkable difference of overall F1-scores between weights with raw frequency and logarithmic scaled frequency. The results where weight is based on logarithmic scaled frequency are higher than those where weight is based on raw frequency, i.e. on average 5.8% higher when using TF-IDF and on average 4.7% higher when using Okapi BM25. These differences are caused by the amplification of exponentiation in logarithmic function. Third, the corpora used to compute the IDF scores also affect the results. With TF-IDF, IDF scores calculated on the legal text corpus yield better performance than on the general text corpus. In contrast, with Okapi BM25, we achieved the best result using the general text corpus.

Error Analysis

Though the best performance is obtained using TF-IDF on legal corpus, the recall is low. Hence, we analyze errors on the setting that produces the second best performance of proposed approach, i.e. using TF-IDF weighting scheme with logarithmic scaled term frequency on the corpus containing the equal number of legal documents and news documents. With this setting, our approach obtains 161 correct keywords out of 464 extracted keywords. In 47 annotated keywords which are not extracted, we found three main reasons:

- Keywords are not wrapped in chunks (4 keywords), e.g. "保険料四分の一免除期間 (One forth of pension fee exemption period)";
- Tokens which are not recognized as independent words are wrongly removed from the candidates (3 keywords), e.g. token "第 (-ary)" and "二 (second)" are removed from candidates "第二号被保険者 (Secondary Insured Person)"; or, unnecessary tokens are remained in candidates(1 keyword), e.g. "等 (etc.)" is remained in candidate "老齢給付等 (Old Age benefits, etc.)" since it is recognized as an independent word;
- Keywords have low scores (39 keywords), e.g. "返還金債権 (claim for refund)", "事務所 (office)", "原因 (cause)", and "手続 (procedure)".

The 303 incorrect extracted keywords are caused by their high weights and the post-processing rules keep irrelevant tokens or remove the relevant tokens from the candidate chunks. For instances

- Keywords “厚生年金保険 (Employees’ Pension Insurance)” and “障害厚生年金 (disability basic pension)” are extracted since they have high weights;
- Token “当該 (relevant)” actually serves grammatical role but it is recognized as an independence word. Hence, it remains in keywords such as ‘当該基金 (funds)”, “当該被保険者 (insured persons)” and “当該傷病 (injuries and diseases).”

3.5.5 Performance Evaluation on Extracting Keyclauses

To the best of our knowledge, there is no work on extracting important Japanese clauses. Therefore, we present the performance of Japanese legal keyclause extraction

Table 3.10: Performance of proposed approach on Japanese legal keyclause extraction on JNPA data.

		#Extr.	**#Corr.**	**Prec.**	**Rec.**	**F1**
Using TF-IDF weighting scheme						
Corpus for IDF	TF					
Law	logTF	1,392	1,061	**76.22**	52.55	62.21
Law	rawTF	1,883	1,401	74.40	69.39	71.81
Mai:’91 to ’95	logTF	2,078	1,521	73.20	75.33	74.25
Mai:’91 to ’95	rawTF	2,212	1,615	73.01	79.99	76.34
Combine all	logTF	2,147	1,576	73.40	78.06	75.66
Combine all	rawTF	2,225	1,622	72.90	80.34	76.44
Law&Mai’95(all)	logTF	2,173	1,591	73.22	78.80	75.91
Law&Mai’95(all)	rawTF	2,232	1,627	72.89	80.58	76.55
Law&Mai’95(part)	logTF	1,881	1,408	74.85	69.74	72.21
Law&Mai’95(part)	rawTF	2,236	1,630	72.90	**80.73**	**76.62**
Using Okapi BM25 weighting scheme: $k1 = 2.0$ and $b = 0.75$						
Corpus for IDF	TF					
Law	logTF	1,802	1,363	75.64	67.51	71.34
Law	rawTF	1,683	1,265	75.16	62.65	68.34
Mai:’91 to ’95	logTF	1,871	1,431	76.48	70.88	73.57
Mai:’91 to ’95	rawTF	1,612	1,260	78.16	62.41	69.40
Combine all	logTF	1,902	1,456	76.55	**72.11**	**74.27**
Combine all	rawTF	1,600	1,261	78.81	62.46	69.69
Law&Mai’95(all)	logTF	1,814	1,412	77.84	69.94	73.68
Law&Mai’95(all)	rawTF	1,502	1,200	**79.89**	59.44	68.16
Law&Mai’95(part)	logTF	1,565	1,191	76.10	58.99	66.46
Law&Mai’95(part)	rawTF	1,276	960	75.24	47.55	58.27

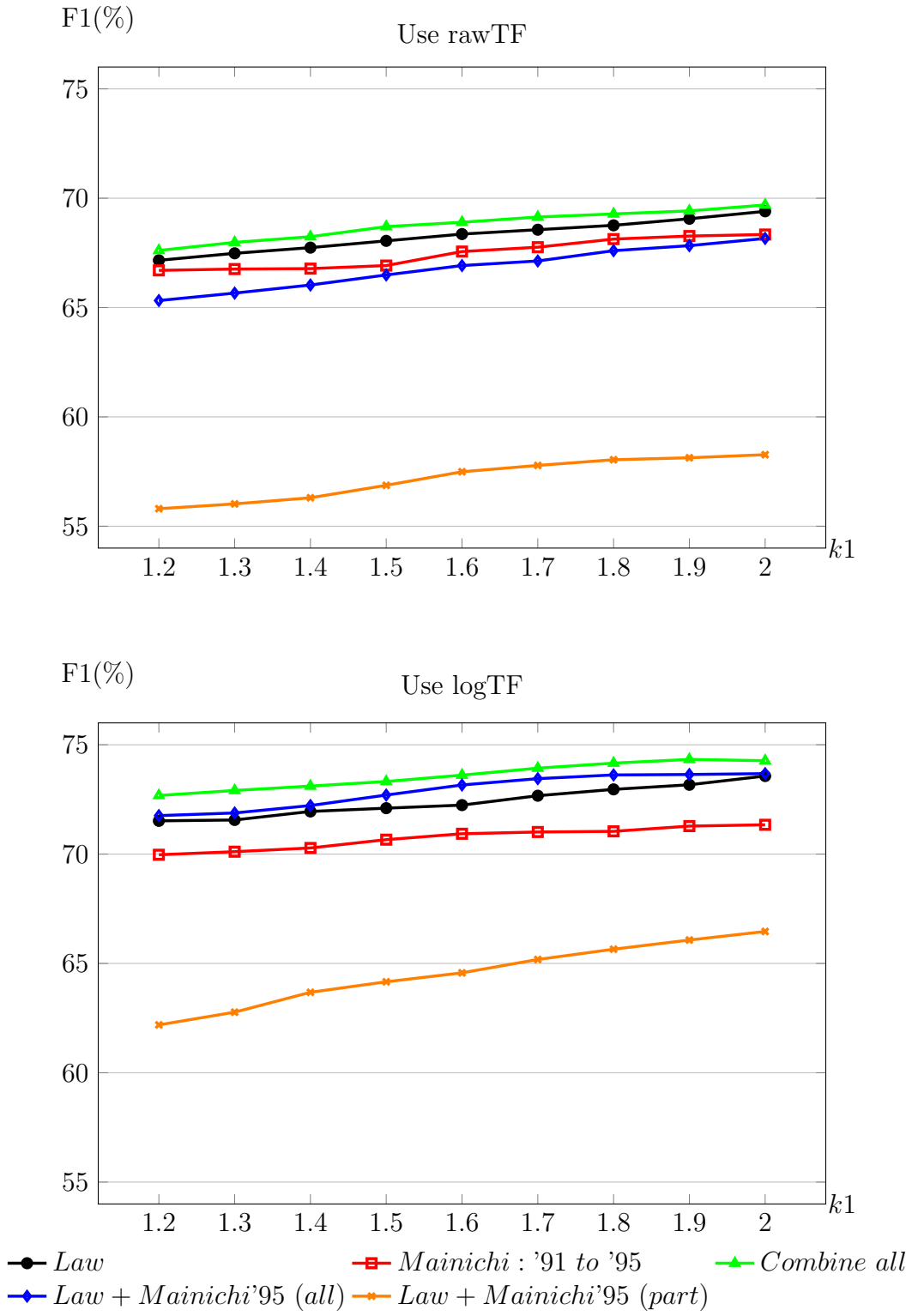


Figure 3.7: The performance of proposed approach on extracting keyclauses from JPNA using Okapi BM25 weighting scheme.

without comparison to another approach. The weighting schemes are TF-IDF and Okapi BM25. The results of keyclause extraction with Okapi BM25 as weighting scheme using different corpus and different values of parameter $k1$ are sketched in Figure 3.7. Similar to extracting keytokens and keyphrases using Okapi, our approach achieves best performances at $k1 = 2.0$ on all corpora and types of term frequencies. The details of performances on keyclause extraction on JNPA data is presented in Table 3.10. In overall, our approach achieves F1-scores at more than 70%. We achieve the best performance at $F1 = 76.62\%$ using TF-IDF and $F1 = 74.27\%$ using Okapi weighting schemes. Therefore, our proposed approach is promising for extracting the important clauses from the legal documents when a clause parser is available.

As analyzed in the previous section, the performances of our approach in extracting keyclauses are also influenced by the weighting schemes, the types of term frequencies and the corpus on which we compute the statistics. First, extracting keyclauses using TF-IDF weighting scheme also result better performances than using Okapi. Second, the performances using Okapi with logarithmic scaled frequency are on average 5.1% higher than that with raw frequency. However, the performance using TF-IDF with logarithmic scaled frequency are on average 3.5% lower than that with raw frequency. Third, corpora on which we compute the statistics for weights of tokens affect the performances. In extracting keyclauses, the best performance using Okapi BM25 is achieved on the largest corpus, but the best performance using TF-IDF is achieved on the corpus with equal number of legal and news documents.

3.6 Conclusions

For knowledge-base of informatics, we proposed a novel unsupervised approach for extracting Japanese indices from legal documents based on Japanese sentence structure. Following an assumption that a legal index is composed of important tokens, we collect Japanese chunks and clauses that contain as many important tokens as possible. The importance of tokens are expressed by weights which are calculated based on the context of a legal document collection. Then, the importance of chunks and clauses are estimated by the average of the weights of tokens in them. The highly weighted chunks and clauses are collected as the legal indices. Although we employ basic NLP concepts and simple techniques, the experimental results shown by F1-scores indicates that our approach is superior to TextRank, the graph-based ranking approach, when extracting keytokens and keyphrases as indices from Japanese legal documents. In addition, we showed a high performance in extracting keyclauses as indices, explaining the influences of different weighting schemes on Japanese legal index extraction.

Chapter 4

English Keyphrase Extraction

This chapter describes the adaption of weight averaging method to extract keyphrases in English text. Current studies often extract keyphrases by collecting adjacent important adjectives and nouns. However, the statistics on four public corpora show that about 15% of keyphrases contain other kinds of words. Even so, incorporating such kinds of words to the noun phrase patterns is not a solution to improve the extraction performance. In this chapter, we describe our solution to improve the extraction performance by involving new kinds of words to keyphrases. First, keyphrase candidates are extracted from noun phrases using syntactic information which is obtained by shallow and deep parsing. Second, candidates are then associated with weights to indicate their importance in documents. The weight of a noun phrase candidate is computed as the average of the weights of tokens in it. Finally, the top weighted candidates in each document are selected as keyphrases for that document. We have experimented on four public corpora to demonstrate that our proposal improve the performance of keyphrase extraction and new kinds of words are introduced to keyphrases. In addition, our proposal is also superior to the current unsupervised keyphrase extraction approaches.

4.1 Introduction

Keyphrases are single-token or multi-token expressions that provide the essential information of a sentence or document. By extracting the keyphrases from documents, it becomes easier for us to obtain the main ideas contained in the documents and to figure out the semantic relations of the contents within and among documents. The extracted keyphrases provide clues which enable us to navigate to related documents or information quickly. Automatic keyphrase extraction plays an important role in many applications of natural language processing (NLP), such as information retrieval, text summarization,

document classification, question answering, and many other applications. However, automatic keyphrase extraction is still a challenge in NLP, especially in the internet era, where the amount of information is continuously increasing. In this situation, manually extracting keyphrases becomes a time-consuming and labor-intensive task.

Many approaches have been proposed for extracting keyphrases automatically. These approaches have two common characteristics: the first is that as many possible tokens that are candidates for keyphrases in documents are collected; the second is that a fixed pattern is applied to collapse potential tokens into keyphrases. Candidates for keyphrases are collected by many methods: applying linguistic knowledge (e.g. syntactic features like part-of-speech tags, NP chunks) and statistics (e.g. term frequency, inverse document frequency, n-grams) as in the works of Turney [Tur99, Tur00], Frank et al. [FPW+99] and Hulth [Hul03]; applying graph-based ranking technique [MT04, WX08b, LLZS09, LL08, BBD13]; or applying clustering technique [MI04, LPLL09]. The fixed pattern for extracting keyphrases is frequently the combination of adjacent candidates which are adjectives and nouns.

Previous research has improved the performance of extraction algorithms by exploring many approaches to enable the collection of as many potential tokens as possible. However, all of them have applied a fixed pattern (a combination of adjectives and nouns which appear adjacently) to decide the form of keyphrases. For this reason, they have restricted candidates to a set of pre-specified words, i.e. nouns and adjectives. Therefore, other kinds of words cannot be selected as candidates, and will consequently never appear in keyphrases.

Practically, not all of keyphrases are composed of adjectives and nouns. Indeed, when shedding a light on the patterns of keyphrases in four corpora, we found that there are approximately 15% of keyphrases contain words other than adjectives and nouns. For examples, some keyphrases which are not composed of only adjectives and nouns are *lower net income*, *nearest parent model*, *partially ordered set*, *category 5 hurricane*, *teaching in IT*, *types of information*, *plug and play methodology*, *ordering criteria*, *waiting time*, *synthesized data*, and *generalized predictive control design*. Among them, roughly 6.5% of keyphrases contain verbs in forms of present and past participles. Since this is a noticeable percentage, we involve these participles to noun phrase patterns when extracting keyphrases. Unfortunately, the extraction performance decreases because the participles which modify noun phrases are confused with the verbs of sentences. By experiments, we have shown that, expanding patterns is not a solution to take into account more kinds of words when extracting keyphrases. However, since a significant percentage of keyphrases contain words other than adjectives and nouns, there should be a way to tackle new kinds of words in keyphrase extraction.

A straightforward solution to take into account more types of words in keyphra-ses is to extract noun phrases from chunks [LNS13]. However, this way does not appropriate to English since English chunks contain tokens, such as punctuation and conjunctions, that may disturb the keyphrase candidates.

We motivate to improve the extraction performance by tackling words other than adjectives and nouns which benefit the performance of keyphrase extraction. In this article, we propose a novel approach to extract the noun phrases as candidate keyphrases using syntactic information, i.e. chunks and constituent syntactic parse tree. Our proposal has four main steps:

1. Collect noun phrases as candidates;
2. Post-process candidates to make sure they are well-formed;
3. Assign weights to candidates to indicate their importance;
4. Rank candidates by descending order of weights and collect the top weighted candidates as the keyphrases.

We experimented keyphrase extraction on four public corpora and achieved very competitive performance. Compare to extraction using patterns and the whole chunkss, our proposal takes advantage in performance while reserving the well-formedness of keyphrases and involving more kinds of words. Compare to the state-of-the-art achievement on each corpus, we beat the state-of-the-art performance on three corpora, but our approach is still behind a supervised approach which employs many features for machine learning. In contrast, our approach exploits syntactic and statistical information to extract keyphrases unsupervisedly. Therefore we are able to conclude that our proposed approach is a competitive approach for unsupervised keyphrase extraction.

The rest of this chapter is organized as follows: Section 4.2 outlines the related approaches of automatic keyphrase extraction; Section 4.3 describes and analyses the corpora for experiments; Section 4.4 explains why the performance decreases when including participles to patterns as well as particularizes our proposal to solve that problem; and Section 4.6 concludes our work in this chapter.

4.2 Related Work

Though many approaches have been proposed for keyphrase extraction, the performances still do not satisfy the expectation and there is room for improvement [KMKB10]. The simplest approach in keyphrase extraction is based on word frequency. However, it

produces unsatisfactory results, which has driven researchers to improve keyphrase extraction approaches in many ways. In general, keyphrase extraction algorithms can be divided into two categories: supervised approaches and unsupervised approaches.

Supervised approaches are mainly based on machine learning to extract keyphr-ases. Keyphrase extraction is usually treated as a classification problem. The basic idea of this approach is to train a classifier to decide if a given word is a keyword or a non-keyword. The researchers have explored many options for training the classifier. Turney [Tur99, Tur00] combines heuristic rules with genetic algorithms to learn the classifier. Frank et al. [FPW+99] alternatively use Naive Bayes method for training. Hulth [Hul03] adds the use of linguistic knowledge like n-grams, part-of-speech tags, and NP chunks to improve performance. Recently, Caragea et al. [CBGG14] have improved the performance of keyphrase extraction for scientific articles by adding citations to feature set when training the classifier. However, the main disadvantage of supervised methods is that the annotated data needed for the training phase is very costly, especially when large amounts of data are required.

Unsupervised approaches for keyphrase extraction employed graph-based ranking algorithms which were first marked by TextRank [MT04]. Mihalcea and Tarau [MT04] have proved that unsupervised methods can work even better than supervised extraction methods. TextRank assumes that a word is important if it has connections to many other words or has connections to other important words. TextRank constructs a graph of words, where the nodes represent the words, and the edges (or relations) between two nodes exist if two words simultaneously appear in a co-occurrence window. After a number of iterations of calculating the node weights, a cutoff threshold is used to select high ranked single nodes as candidates. Finally, these candidates are collapsed to become keyphrases.

Inspired by the idea of TextRank, a number of variants of graph-based ranking algorithms has been proposed, such as adding weight to edges, keeping all vertices of the graph and expanding the co-occurrence window in SingleRank [WX08b]; exploiting the adjacent knowledge to extract keyphrases in ExpandRank [WX08b]; and employing the clustering exemplars in ranking [WX08a]. Liu et al. [LHZS10], in Topical PageRank, combine the ranking approach with knowledge obtained from topic modeling. Litvak and Last [LL08] take into account the in-degree and out-degree nodes, as well as point out that the ranking approach can be converged by one iteration. Negi [Neg14] extends TextTank and takes into account the document metadata labels to extract label-specific keyphrases from a multi-labeled document. The ranking-based approach is also widely applied in extracting keyphrases from the web. For example, Wan and Xiao [WX08a], in CollabRank algorithm, combine document clusters to graph for ranking; Wu et al. [WZO10] employ

TextRank to extract annotation tags from Twitter; Zhao et al. [ZJH+11] improve Topical PageRank [LHZS10] by adding topic-sensitive score propagation to extract keyphrases from Twitter.

Another notable unsupervised approach is the clustering-based method [MI04, LPLL09]. Liu et al. [LPLL09] cluster words based on semantic distances (Wikipedia-based statistic) of single words to obtain cluster exemplars that represent the content of a given document. These exemplars are then used as the clues to extract keyphrases that include one or more exemplars. Hasan and Ng [HN10] have confirmed that clustering-based approaches perform better than graph-based ranking approaches on many data sets. Recently, Bougouin et al. [BBD13] apply clustering to discover the topics of documents and integrate those topics into graph to run ranking for keyphrases. Le et al. [LNS13] collect highly weighted chunks to extract keyphrases from Japanese legal documents. An broader overview of approaches for automatic keyphrase extraction can be found in a survey by Hasan and Ng [HN14].

4.3 Corpora and Keyphrase Analysis

Hasan and Ng [HN10] have demonstrated that the performance of keyphrase extraction should be evaluated on different data sets. Therefore, we consider four public corpora, namely DUC-2001, Inspec, NUS, and SemEval-2010, which are used for evaluating the extraction performance in previous studies.

DUC-2001 corpus [Ove01] includes 593 news articles dividing into two sets: 285 articles for training and 308 articles for testing. This corpus has been annotated by Wan and Xiao [WX08b]. Since this corpus is stored in XML format, we pre-process for the text and remove metadata XML tags: `DOCNO`, `PROFILE`, `DOCID`, `DATE`, `FILEID`, `FIRST`, `SECOND`, `LENGTH`, `NOTE`, and `ACCESS`. The contents of all 308 articles are used for keyphrase extraction.

Inspec corpus [Hul03] includes 2,000 abstracts from journal articles in discipline of *Computers and Control* and *Information Technology*. This corpus is divided into three sets: 1,000 abstracts for training, 500 abstracts for development and 500 abstracts for testing. We use 500 abstracts in test set for evaluation.

NUS corpus [NK07] includes 211 full scientific conference papers. Since the keyphrases frequently appear in abstracts, we use only titles and abstracts of 211 conference papers to extract keyphrases.

SemEval-2010 corpus [KMKB10] includes 244 conference and workshop papers splitting to two sets: 144 papers for training and 100 papers for testing. Only titles and abstracts of 100 papers in test set are used for evaluation.

Table 4.1: The characteristics of four public corpora of keyphrase extraction.

	Corpora			
	DUC-2001	**Inspec**	**NUS**	**SemEval-2010**
Type	News articles	Paper abstracts	Paper abstracts	Paper abstracts
# Documents for test	308	500	211	100
# Keys	2,484	4,913	2,327	1,482
# One-word keys	431 (17.4%)	659 (13.4%)	610 (26.2%)	309 (20.9%)
# Keys (adj+noun)	2,298 (92.5%)	4,221 (85.9%)	1,903 (81.8%)	(84.5%)
# Keys (w. participles)	53 (2.1%)	383 (7.8%)	206 (8.9%)	(7.2%)
# Keys (other patterns)	133 (5.4%)	309 (6.3%)	218 (9.4%)	(8.3%)
# Exist. keys	2,462 (99.1%)	3,826 (77.9%)	2,200 (94.5%)	(89.5%)
# Exist. keys (adj+noun)	2,277 (91.7%)	3,338 (68%)	1,837 (78.9%)	(80%)
# Exist. keys (w. participles)	53 (2.1%)	287 (5.8%)	178 (7.6%)	(3.2%)
# Exist. keys (other patterns)	132 (5.3%)	201 (4.1%)	185 (8%)	(6.3%)

Some characteristics of these corpora have been analyzed in previous studies [HN10, KMKB10]. In this study, we examine two other characteristics in concern of our work and show them in Table 4.1. The characteristics in our concern are: the percentage of one-word keyphrases and the percentage of the types of keyphrase patterns. Each corpus has a different percentage of one-word keyphrases and the percentage of one-word keyphrases on four corpora is 19.5% on average. Since this is a significant percentage, a certain percentage of one-word keyphrases should be specified when extracting keyphrases.

When analyzing the patterns of keyphrases, we observe that not only adjectives and nouns appear in keyphrases, but other types of words also appear, such as:

- Verbs in forms of present and past participles which serves grammatical roles as nouns and adjectives, e.g. *watermarking*, *ordering criteria*, *synthesized data*, and *user defined virtual collections*;
- Adjectives in forms of comparative and superlative, e.g. *higher education*, *lower net income* and *nearest parent model*;
- Adverbs, e.g. *highly nonlinear rule-based models*, *visually impaired people* and *partially ordered set*;
- Cardinal numbers, e.g. *four main design patterns*, *category 5 hurricane* and *type II diabetes*;
- Conjunctions, e.g. *plug and play methodology*, *Security and Privacy* and *training vs. education*;

- Other kinds of words such as prepositions or subordinating conjunctions (e.g. *teaching in IT*, *types of information*, *quality of service*, *payoff per share*, *design to cost system*, and *image processing with crayons*); apostrophes (e.g. *rebel new people's army* and *pizarro's assassination*); and foreign words (e.g. *ad hoc networks* and *basic Japanese kanji*).

The percentage of keyphrases for each type of keyphrase pattern is showed next to the number of keyphrases in Table 4.1. Note that, keyphrases in test set of SemEval-2010 are provided as stemmed words, we examine the characteristic of keyphrases in training set instead. As shown in Table 4.1, the percentage of keyphrases which follow the patterns of adjectives and nouns is 86.2% on average; the percentage of keyphrases which contain verbs in forms of present and past participles is 6.5% on average; and percentage of the other patterns of keyphrases is 7.3% on average.

When looking closely to the annotated keyphrases, about 10% of them actually do not appear in the text. Among 90% of existing keyphrases, the percentage of keyphrases which follow the patterns of adjectives and nouns is 79.6% on average; the percentage of keyphrases which contain participles is 4.7% on average; and percentage of the other patterns of keyphrases is 5.9% on average. These percentages explicit that when involving only adjectives and nouns, the highest recall of extraction performance is less than 80%.

Among other types of words in keyphrases, verbs in forms of present and past participles have a considerable contribution to keyphrases. Therefore, we examine whether involving participles as candidates in keyphrase patterns improves the extraction performance. The experimental results in the following section demonstrate that involving participles to keyphrase patterns decreases the performance of keyphrase extraction since the participles are probably confused with the verbs of sentences. To tackles words other than adjectives and nouns to keyphrase, we propose an approach which employs constituent parse tree.

4.4 Keyphrase Extraction with Average TF-IDF Scores

Keyphrases are expressions which point out the main ideas of sentences or documents. A keyphrase can be either a *single token* or a *multi-token* expression. For convenience, in specific contexts, we refer to a single token keyphrase as a *single keyphrase* and a multi-token keyphrase as a *compound keyphrase*. In general contexts, we refer to the term *keyphrase*.

Since a majority of keyphrases are nouns and noun phrases [Hul03], we extract nouns and noun phrases as candidates for keyphrases. Previous approaches often extract adjec-

tives and nouns, which benefit keyphrase extraction, as candidates and collapsing them to form keyphrases. Instead of collecting individual tokens, we firstly collect noun phrases as candidates for keyphrases. Then, the most important candidates in a document are selected as keyprhases. In brief, the outline of all techniques presented in this study contains four main steps:

1. Extract candidates which are noun phrases;
2. Post-process candidates;
3. Assign a weight to each candidate;
4. Collect the top weighted candidates as keyphrases.

The details of extracting noun phrases and and post-processing candidates are described in Section 4.5.

A noun phrase is extracted as a keyphrase of a document if it is determined as important in that document. Following an assumption that keyphrases are composed of important tokens [LNS13], the weight of a noun phrase NP is decided as the average weight of tokens which are included in that phrase:

$$weight_{NP} = \frac{\sum_{t \in NP} weight_t}{|NP|}$$

where, $|NP|$ is the number of tokens and $weight_t$ is the TF-IDF score of a token t in noun phrase. Note that, when computing this average, we do not count the tokens whose weights are zero. TF-IDF (Term Frequency - Inverse Document Frequency) score indicates the importance of a token and is computed as the product of TF and IDF scores:

$$tfidf_{t,d} = tf_{t,d} \times idf_{t,D}$$

In which, the IDF (Inverse Document Frequency) of a token t (denoted by $idf_{t,D}$) is a score which indicates the importance of that token in a collection of documents D:

$$idf_{t,D} = \log \frac{N}{df_t}$$

The TF (Term Frequency) of a token t is the raw frequency of token t (denoted by f_t) in a document d or the logarithmic scaled of raw frequency:

$$tf_{t,d} = \log (f_t + 1)$$

In experiments, all documents in each corpus are used to calculate IDF scores of tokens. Specifically, the numbers of documents on which IDF scores of tokens are computed are respectively 593 for DUC-2001, 2,000 for Inspec, 211 for NUS, and 244 for SemEval-2010. The weights of tokens which show stop-words or punctuation are reset to zero. Note that the stop-words list is reused from a work by Salton [Sal71].

4.5 Extracting Candidates for Keyphrases

In this section, we describe three ways to extract noun phrase candidates for keyphrase extraction using patterns, chunks and parse trees. Pattern is a traditional way to extract noun phrase candidates. Though keyphrase are often the combinations of adjectives and nouns, keyphrases actually contain a noticeable percentage of participles. Unfortunately, incorporating such participles to keyphrase patterns also cause the noise since the participles are confused with the main verbs of sentences. Therefore, we try to extract noun phrase chunks as candidates when extracting keyphrases, yet this technique introduces another noise because of the punctuation and conjunctions. In this research, we introduce a novel technique improve the performance of keyphrase extraction by exploiting the syntactic information, i.e. constituent parse trees and chunks.

The following parts of this section is organized as follows: Section 4.5.1 explains how to extract keyphrases using patterns; Section 4.5.2 describes the keyphrase extraction using the whole chunks from text; and Section 4.5.3 introduces our proposal to tackle more kinds of words in keyphrase extraction.

4.5.1 Keyphrase Extraction using Patterns of Noun Phrases

This section describes the extracting process using patterns of noun phrases. We employ Stanford CoreNLP tool[1], which uses Penn Tree Bank POS tag set, to annotate the text. Since approximately 86% of keyphrases are combinations of adjectives and nouns, to collect noun phrases, previous work often uses a pattern to collapse adjacent adjectives and nouns. When examine the keyphrases pattern, we recognize that adjectives in forms of comparative and superlative also appear, e.g. *lower net income* and *nearest parent model*. Hence, the pattern for noun phrases is modified as following regular expression

$$(JJ|JJR|JJS)*(NN|NNS|NNP|NNPS)+$$

As analyzed in Section 4.3, on average 6.5% of verbs in forms of present and past participles play the roles as adjectives and nouns in keyphrases. Hence, we involve them

[1]Stanford CoreNLP API is available at `http://nlp.stanford.edu/software/corenlp.shtml`

to the pattern of noun phrases and introduce another pattern for noun phrases to examine whether including such participles improves extraction performance as following

$$(JJ|JJR|JJS|VBG|VBN)*(NN|NNS|NNP|NNPS|VBG)+$$

Experiments are run on four public corpora described in Section 4.3. Following the evaluation criteria in SemEval-2010, we extract up to 15 highest weighted keyphrases for each document in which single keyphrases and compound keyphrases are extracted separately. For each document, we collect 1 single keyphrases and 14 compound keyphrases. The extracted and annotated keyphrases are stemmed using the English Porter stemmer[2] when counting for performance. An extracted keyphrase is counted as correct if its stemmed keyphrase is exactly matched with a stemmed annotation.

This technique is compared to a baseline, henceforth referred as *TF-IDF n-grams* for convenience. In TF-IDF n-grams, the top weighted *n*-grams of adjectives and nouns are extracted as keyphrases where the weight of a candidate is calculated by summing its constituent unigrams. The performances of this baseline have been reported in previous works [HN10, KMKB10].

The performance of the proposed technique is presented in Table 4.2. Our technique achieves better performance than the TF-IDF n-grams baseline for all corpora. Henceforth, we use these results as new baseline for keyphrase extraction in this work. When adding verbs in forms of present and past participles to the pattern of keyphrases, the performance decreases. The reason is that the participles which modify the meaning of noun phrases are confused with the verbs of sentences. For an example, the phrase *indicated differing levels* in the following sentence is wrongly extracted as a keyphrase since it satisfies the pattern of keyphrases.

*"Previous research has **indicated differing levels** of importance of perceived ease of use relative to other factors."*

In fact, the participle *indicated* does not modify the meaning of noun phrase *differing levels* but it is a conjugation of verb in present participle tense.

Based on experimental results, we conclude that the performance of keyphrase extraction is not improved when involving present and past participles into noun phrase patterns. However, as keyphrasess contain such parts-of-speech of words, an approach should be investigated to capture all possible words to keyphrases.

[2]Java implementation for English Porter stemmer is available at `http://tartarus.org/martin/PorterStemmer/`

Table 4.2: The performance of English keyphrase extraction using patterns.

		#Extr.	#Corr.	Prec.	Rec.	F1
DUC-2001						
Pattern (adj+noun)	logTF	4,620	925	20.0	37.2	26.0
	rawTF	4,619	984	21.3	39.6	**27.7**
Pattern (+participle)	logTF	4,620	906	19.6	36.5	25.5
	rawTF	4,619	963	20.8	38.8	27.1
TF-IDF *n*-grams		-	-	-	-	27.0
Inspec						
Pattern (adj+noun)	logTF	6,232	2,224	35.7	45.3	**39.9**
	rawTF	6,232	2,204	35.4	44.9	39.6
Pattern (+participle)	logTF	6,445	2,223	34.5	45.2	39.1
	rawTF	6,445	2,193	34.0	44.6	38.6
TF-IDF *n*-grams		-	-	-	-	36.3
NUS						
Pattern (adj+noun)	logTF	2,995	461	15.4	19.8	17.3
	rawTF	2,995	471	15.7	20.2	**17.7**
Pattern (+participle)	logTF	3,056	443	14.5	19.0	16.5
	rawTF	3,056	455	14.9	19.6	16.9
TF-IDF *n*-grams		-	-	-	-	6.6
SemEval-2010						
Pattern (adj+noun)	logTF	1,465	308	21.0	20.8	**20.9**
	rawTF	1,465	307	21.0	20.7	20.8
Pattern (+participle)	logTF	1,484	294	19.8	19.8	19.8
	rawTF	1,484	288	19.4	19.4	19.4
TF-IDF *n*-grams		-	-	14.9	15.3	15.1

4.5.2 Keyphrase Extraction using Noun Phrases in Chunks

A chunk is a group of words which has a specific meaning. The problem of *chunking*, so called *shallow parsing*, is a problem of separating natural language sentences into phrases such as noun phrases and verb phrases. [LNS13] have extracted the noun phrases which are wrapped in chunks for Japanese. Hence, we apply this technique on English text to examine whether the extraction performance is improved when collecting noun phrases from chunks as candidates. First, each sentence in document is parsed for chunks using Illinois Chunker[3] [PR00].

[3]Java API for Illinois Chunker is available at `http://cogcomp.cs.illinois.edu/page/software_view/Chunker`

[NP (DT This) (NN article)] [VP (VBZ provides)] [NP (JJ detailed) (NN advice)] [PP (IN on)] [VP (VBG acquiring)] [NP (JJ new) (, ,) (JJ out-of-print) (, ,) (CC and) (JJ rare) (NNS materials)] [PP (IN in)] [NP (DT the) (NN history)] [PP (IN of)] [NP (NN science)] (, ,) [NP (NN technology)] (, ,) (CC and) [NP (NN medicine)] [PP (IN for)] [NP (DT the) (NN beginner)] [PP (IN in)] [NP (DT these) (NNS fields)] (. .)

Figure 4.1: An example of chunks in an English sentence.

Figure 4.1 shows an example of chunks in an English sentence, in which noun phrase and verb phrase chunks are coloured. Nouns and noun phrases which are wrapped in chunks are collected as candidates for keyphrases. These candidates are post-processed by removing the tokens, whose weight are zero, from the beginning or ending of candidates. In this work, we introduce an additional post-processing step to ensure that candidates are well-formed. This step removes the unnecessary tokens from the beginning and ending of candidates. Two ways are introduced to remove unnecessary tokens to ensure that:

- A Candidate begins with a token whose POS tag is JJ, JJR, JJS, NN, NNS, NNP, or NNPS; and ends with a token whose POS tag is NN, NNS, NNP, or NNPS.
- A Candidate begins with a token whose POS tag is JJ, JJR, JJS, NN, NNS, NNP, NNPS, VBG, or VBN; and ends with a token whose POS tag is NN, NNS, NNP, NNPS, or VBG.

In other words, we consider only adjectives and nouns in the first way while involving participles to the candidates in the second way. Finally, the top weighted candidates in a document are selected as keyphrases for that document.

We applied this technique on four corpora described in Section 4.3 and used the same evaluation criteria. The extraction results are shown in Table 4.3. Note that, performance of keyphrase extractions using patterns of adjectives and nouns is used as baseline for chunk based keyphrase extraction.

As shown in Table 4.3, the extraction performance is improved in only one corpus, i.e. SemEval-2010 when using chunks to collect candidates. The performance on other corpora is lower than the baseline. The reason of is that noun phrases, which are obtained by shallow parsing, contain punctuation and conjunctions. Such tokens rarely appear in keyphrases, they therefore cause a detriment to keyphrase extraction. For example, the keyphrase *rare materials* is missed while noun phrase *new, out-of-print, and rare materials* is wrongly extracted as a keyphrase from the example sentence in Figure 4.1. From the experimental results, we found that, when extracting chunks as candidates, involving participles in chunks has slightly higher recall than using adjectives and nouns. However,

Table 4.3: The performance of English keyphrase extraction using chunks.

		#Extr.	#Corr.	Prec.	Rec.	F1
DUC-2001						
Chunk	logTF	4,615	874	18.9	35.2	24.6
(adj+noun)	rawTF	4,616	939	20.3	37.8	26.5
Chunk	logTF	4,616	869	18.8	35.0	24.5
(+participle)	rawTF	4,617	946	20.5	38.1	26.6
Pattern	logTF	4,620	925	20.0	37.2	26.0
(adj+noun)	rawTF	4,619	984	21.3	39.6	**27.7**
Inspec						
Chunk	logTF	5,924	2,066	34.9	42.1	38.1
(adj+noun)	rawTF	5,924	2,032	34.3	41.4	37.5
Chunk	logTF	6,061	2,095	34.6	42.6	38.2
(+participle)	rawTF	6,061	2,070	34.2	42.1	37.7
Pattern	logTF	6,232	2,224	35.7	45.3	**39.9**
(adj+noun)	rawTF	6,232	2,204	35.4	44.9	39.6
NUS						
Chunk	logTF	2,917	447	15.3	19.2	17.0
(adj+noun)	rawTF	2,917	459	15.7	19.7	17.5
Chunk	logTF	2,977	447	15.0	19.2	16.9
(+participle)	rawTF	2,977	462	15.5	19.9	17.4
Pattern	logTF	2,995	461	15.4	19.8	17.3
(adj+noun)	rawTF	2,995	471	15.7	20.2	**17.7**
SemEval-2010						
Chunk	logTF	1,424	303	21.3	20.4	20.9
(adj+noun)	rawTF	1,424	301	21.1	20.3	20.7
Chunk	logTF	1,454	311	21.4	21.0	21.2
(+participle)	rawTF	1,454	317	21.8	21.4	**21.6**
Pattern	logTF	1,465	308	21.0	20.8	20.9
(adj+noun)	rawTF	1,465	307	21.0	20.7	20.8

compare to the baseline, using the whole chunks as candidates does not appropriate to extracting keyphrases from English text.

4.5.3 Keyphrase Extraction using Syntactic Information

In this part, we described our proposal to take into account participles when extracting keyphrase candidates using syntactic information which is resulted by parsing. We use

two levels of parsing for keyphrase extraction: shallow parsing (chunk) and deep parsing (constituent parse tree).

The shallow parsing (chunk) has been described in Section 4.5.2 where an example of chunks in an English sentence is shown in Figure 4.2. The constituent parse tree of

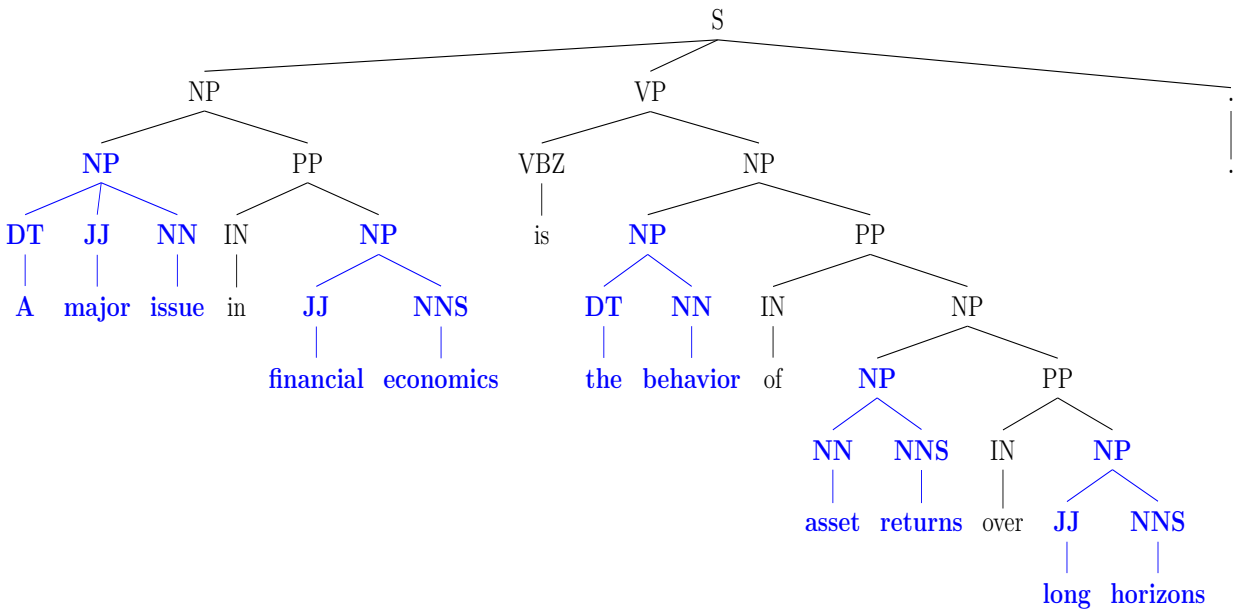


Figure 4.2: An example of parse tree for an English sentence.

a sentence is a tree that represents the syntactic structures of the sentence according to context-free grammars. In the parse tree, the interior nodes and leaf nodes are called *non-terminals* and *terminals* respectively. Figure 4.2 gives an example parse tree in which a sentence is represented as a derivation from the root node S to non-terminals of a noun phrase (NP) and a verb phrase (VP) until reaching the terminals containing the particular words (tokens). In other words, a terminal is a leaf node and shows a particular word; a non-terminal is a branch node that is decomposed to either terminals or non-terminals.

Our proposal to extract keyphrases using syntactic information is outlined in Algorithm 2.

Illinois Chunker and Stanford CoreNLP tools are respectively employed to parse sentences into chunks and parse trees. Then, noun phrases are extracted using these syntactic information. Using shallow parsing, noun phrases are chunks whose tags are NP. Using deep parsing, noun phrases are extracted from non-terminals which are tagged as NP and do not contain other non-terminals. The coloured noun phrases in Figure 4.1 and Figure 4.2 are examples of noun phrases being extracted using syntactic information. After that, each noun phrase is post-processed to eliminate the punctuation, conjunctions and unnecessary tokens. In post-processing, each noun phrase is split at the position of punctuation

Algorithm 2: Keyphrase Extraction using Syntactic Information

input : A document d
output: Set of keyphrases K

```
Parse all sentences in document d for syntactic information;
N ← Noun phrases from document d;
Set of keyphrase candidate C ← ∅;
foreach noun phrase n ∈ N do
    if n contains punctuation or conjunctions then
        N' ← Split(n);
        N ← N ∪ N';
    else
        n' ← RemoveUnnecessaryTokens(n);
        Assign a weight to n';
        C ← C ∪ n';
end
Rank all keyphrase candidates in C by descending order of weights;
K ← Top weighted candidates in C;
```

or conjunctions (if any). The removal of unnecessary tokens from beginning and ending is the same with the process introduced in Section 4.5.2.

The experiments of our proposal are also run on the same data with previous techniques. The details of experimental results are shown in Table 4.4. The extraction performance using syntactic information is shown in Table 4.5 in comparison to other approaches.

We compare our approach to three baselines:

- Pattern baseline: the performance of keyphrase extraction using patterns of adjectives and nouns, which has been described in Section 4.5.1;
- Chunk baseline: the performance of keyphrase of extraction using the whole of English chunks, which has been described in Section 4.5.2;
- Previous best F1 baseline: the state-of-the-art performance on each corpus which has been reported in previous works. For DUC-2001 and NUS corpora, TFDIF n-grams yields the state-of-the-art performance [HN10]. For Inspec corpus, clustering approach [LPLL09] achieves highest F1-score. For SemEval-2010 corpus, HUMB [LR10], a supervised system, obtains the best performance.

The comparisons in Table 4.5 show that, in all corpora, our proposed approach beats the performance of Pattern and Chunk baselines. In most of cases, the precision and recall scores are higher than these two baselines. When comparing to the previous best

Table 4.4: The details of English keyprhase extraction using syntactic information.

		#Extr.	#Corr.	Prec.	Rec.	F1
DUC-2001						
Chunk	logTF	4,614	921	20.0	37.1	26.0
(adj+noun)	rawTF	4,615	984	21.3	39.6	27.7
Chunk	logTF	4,615	915	19.8	36.8	25.8
(+participle)	rawTF	4,616	989	21.4	39.8	27.9
Parse tree	logTF	4,615	916	19.8	36.9	25.8
(adj+noun)	rawTF	4,615	976	21.1	39.3	27.5
Parse tree	logTF	4,615	913	19.8	36.8	25.7
(+participle)	rawTF	4,615	975	21.1	39.3	27.5
Inspec						
Chunk	logTF	5,775	2,258	39.1	46.0	42.3
(adj+noun)	rawTF	5,775	2,227	38.6	45.3	41.7
Chunk	logTF	5,940	2,291	38.6	46.6	42.2
(+participle)	rawTF	5,940	2,264	38.1	46.1	41.7
Parse tree	logTF	5,720	2,217	38.8	45.1	41.7
(adj+noun)	rawTF	5,720	2,196	38.4	44.7	41.3
Parse tree	logTF	5,860	2,227	38.0	45.3	41.3
(+participle)	rawTF	5,860	2,201	37.6	44.8	40.9
NUS						
Chunk	logTF	2,883	482	16.7	20.7	18.5
(adj+noun)	rawTF	2,883	505	17.5	21.7	19.4
Chunk	logTF	2,953	481	16.3	20.7	18.2
(+participle)	rawTF	2,953	507	17.2	21.8	19.2
Parse tree	logTF	2,860	475	16.6	20.4	18.3
(adj+noun)	rawTF	2,859	487	17.0	20.9	18.8
Parse tree	logTF	2,907	473	16.3	20.3	18.1
(+participle)	rawTF	2,906	488	16.8	21.0	18.7
SemEval-2010						
Chunk	logTF	1,404	322	22.9	21.7	22.3
(adj+noun)	rawTF	1,404	319	22.7	21.5	22.1
Chunk	logTF	1,439	333	23.1	22.5	22.8
(+participle)	rawTF	1,439	337	23.4	22.7	23.1
Parse tree	logTF	1,402	319	22.8	21.5	22.1
(adj+noun)	rawTF	1,401	315	22.5	21.3	21.9
Parse tree	logTF	1,427	321	22.5	21.7	22.1
(+participle)	rawTF	1,426	312	21.9	21.1	21.5

Table 4.5: The performance of keyphrase extraction using syntactic information in comparison to other approaches.

		DUC-2001			Inspec			NUS			SemEval-2010		
		Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
Chunk(split)	logTF	20.0	37.1	26.0	39.1	46.0	**42.3**	16.7	20.7	18.5	22.9	21.7	22.3
(adj+noun)	rawTF	21.3	39.6	27.7	38.6	45.3	41.7	17.5	21.7	**19.4**	22.7	21.5	22.1
Chunk(split)	logTF	19.8	36.8	25.8	38.6	46.6	42.2	16.3	20.7	18.2	23.1	22.5	22.8
(+participle)	rawTF	21.4	39.8	**27.9**	38.1	46.1	41.7	17.2	21.8	19.2	23.4	22.7	**23.1**
Parse tree	logTF	19.8	36.9	25.8	38.8	45.1	41.7	16.6	20.4	18.3	22.8	21.5	22.1
(adj+noun)	rawTF	21.1	39.3	27.5	38.4	44.7	41.3	17.0	20.9	18.8	22.5	21.3	21.9
Parse tree	logTF	19.8	36.8	25.7	38.0	45.3	41.3	16.3	20.3	18.1	22.5	21.7	22.1
(+participle)	rawTF	21.1	39.3	27.5	37.6	44.8	40.9	16.8	21.0	18.7	21.9	21.1	21.5
Chunk	logTF	18.9	35.2	24.6	34.9	42.1	38.1	15.3	19.2	17.0	21.3	20.4	20.9
(adj+noun)	rawTF	20.3	37.8	26.5	34.3	41.4	37.5	15.7	19.7	17.5	21.1	20.3	20.7
Chunk	logTF	18.8	35.0	24.5	34.6	42.6	38.2	15.0	19.2	16.9	21.4	21.0	21.2
(+participle)	rawTF	20.5	38.1	26.6	34.2	42.1	37.7	15.5	19.9	17.4	21.8	21.4	21.6
Pattern	logTF	20.0	37.2	26.0	35.7	45.3	39.9	15.4	19.8	17.3	21.0	20.8	20.9
(adj+noun)	rawTF	21.3	39.6	27.7	35.4	44.9	39.6	15.7	20.2	17.7	21.0	20.7	20.8
Previous best F1		-	-	27.0	-	-	40.6	-	-	6.6	27.2	27.8	**27.5**

F1 scores, our proposal achieves the best performance on three corpora: DUC-2001, Inspec and NUS. Note that, the performance on Inspec corpus achieved by clustering approach is reported as 45.7% of F1 score since the evaluation is counted on the keyphrases which appear in the documents. If we also count on the existing keyphrases, we achieve 47.1% of F1 as our performance on Inspec. On SemEval-2010 corpus, our approach still behind HUMB because this is a supervised method which exploits many features for machine learning: structure of the article, lexical cohesion of a sequence of words, TF-IDF scores, and the frequency of the keyword in the global corpus.

When using the syntactic information for extracting candidates, we found that the recall is generally higher if participles are taken into account. In addition, other kinds of words, e.g. cardinal numbers, which occur in the middle of keyphrases are also included. For example, keyphrases *modulo 2 residue class*, *category 5 hurricane* and *type II diabetes* are extracted by using syntactic information no matter participles are tackled or not. Even though words other than adjectives and nouns are involved, the syntactic information keeps the well-formedness of keyphrases. Therefore, both the recall and precision increase.

4.6 Conclusions

In this chapter, we have demonstrated that keyphrases are not consistently the combination of adjectives and nouns. There are roughly 15% of keyphrases including other kinds of words such as participles, comparative/superlative adjectives and cardinal numbers. We believe that participles should be considered in keyphrase extraction since there is a recognizable percentage of keyphrases containing participles (6.5%). However, involving participles to keyphrase patterns even decrease the performance because of the confusion with the main verbs of sentences. To improve the extraction performance and to take into account new kinds of words in keyphrases, we proposed to incorporate the syntactic information when extracting noun phrases as keyphrase candidates. As expected, the experimental results on four public corpora has been improved and new kinds of words has been also introduced to the keyphrases.

Chapter 5

Constructing Hierarchy of Legal Indices

The idea of hierarchical index is applied to the legal domain to provide the readers a general understanding of legal concepts via their super/sub-ordinate relations. This work serves as effort in discovering relationships among legal concepts for automatic legal ontology learning, in which super/sub-ordinate relations are considered. Indices are extracted from legal documents as keywords and their relationships are discovered by language processing method. We propose an approach to extract the super/sub-ordinate relation between each pair of concepts individually based on directional similarity. The relations among a set of legal indices are represented in a directed graph and the hierarchical structure of indices is simply exported from this graph. We adopt this proposal to the Japanese National Pension Act document. The resulted hierarchical structure is compared to an annotated legal ontology on the number of correct relations. The proposed method achieves 40.6% for precision, 46.9% for recall and 43.5% for F-measure as the performance.

5.1 Introduction

The system of legal documents plays an important role in all countries and organizations in ruling the society. Legal documents include: *i*) the documents such as constitution, laws, rules, codes, ordinances or acts, which are promulgated by the government or the administrative organizations; *ii*) the documents recording legal activities such as congresses or courts; *iii*) the civil documents such as contracts, agreements or wills, which are officially composed by lawyers. The scope of this work considers only legal documents which are published by the government or administrative organizations.

Generally, the system of legal documents has three main characteristics:

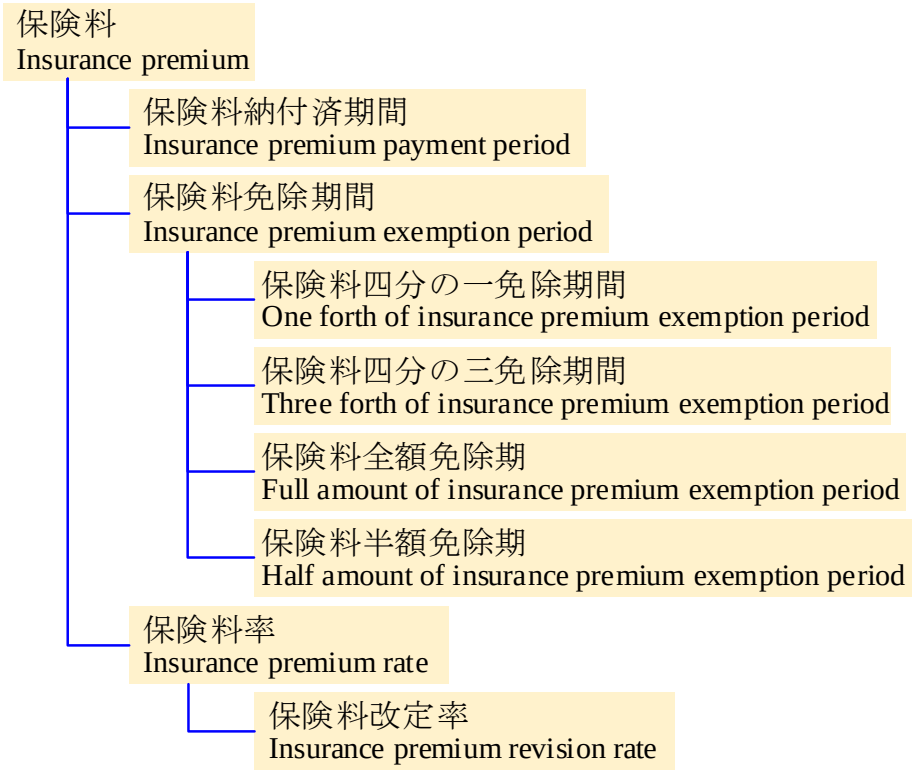
(i) The system of legal documents includes a vast number of documents to described structures and procedures in specific domains of our society.

(ii) Legal documents are frequently updated to keep up with the changes of our society.

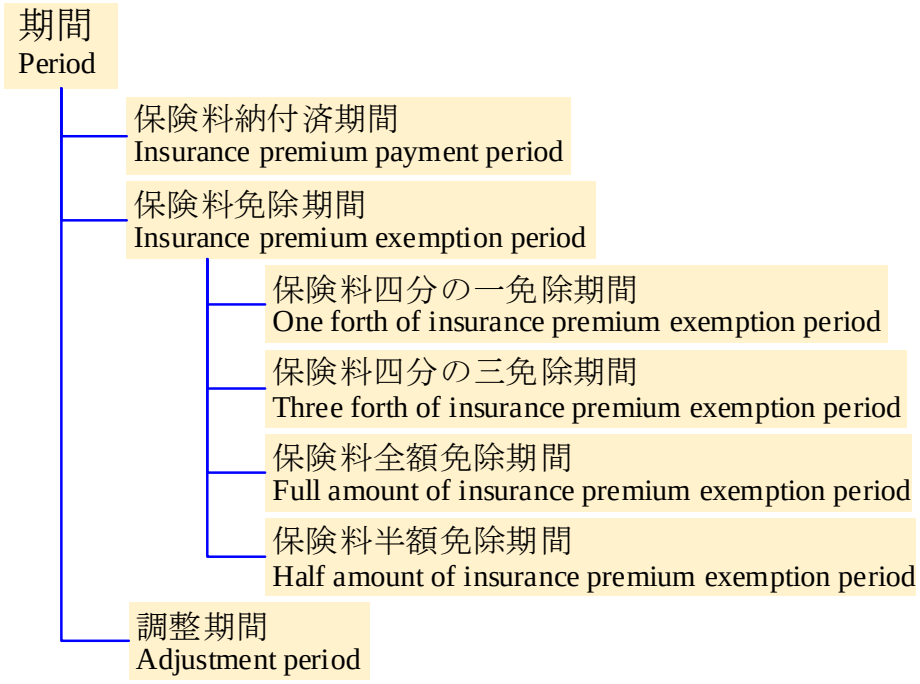
(iii) Each legal document is composed by long complicated sentences which describe how things work, relying upon other parts in the same document and other documents.

Due to the information load and the complexity of their content, managing and understanding the legal documents are difficult for both normal citizens and legislators. For this reason, we aim to support them in reading and understanding these legal documents by providing a briefly structured representation of the legal concepts. This work contributes to the research field of Legal Engineering [Kat07] which focuses on methodology to make, analyze and maintain legal documents as well as methodology to develop law-based information systems.

This study addresses the problem of representing relations among legal concepts, specifically, super-ordinate and sub-ordinate relations among general concepts and specific concepts are considered. We borrow the idea of the hierarchical index of a book to represent the legal concepts hierarchically. The hierarchical index of legal documents is a tree view structure representing the super/sub-ordinate relations among the legal concepts, in which the legal concepts of indices at lower levels are more detailed than the concepts of indices at higher levels. Figure 5.1 illustrates two examples of hierarchical index for some legal concepts from the Japanese National Pension Act. In these hierarchies, the concepts in lower levels are more specific than those in higher levels in meaning. Specifically, the hierarchy in Figure 5.1a expresses the super-ordinate relations of the index 保険料 *(insurance premium)* to its descendants 保険料納付済期間 *(insurance premium payment period)*, 保険料免除期間 *(insurance premium exemption period)* and 保険料率 *(insurance premium rate)*; in turn, index 保険料率 *(insurance premium rate)* descends to more specific index 保険料改定率 *(insurance premium revision rate)*. Similarly, the index 保険料改定率 *(insurance premium revision rate)* has sub-ordinate relation to the index 保険料率 *(insurance premium rate)*. This hierarchical structure is a concept classifier as one general concept may have many sub-ordinate concepts which extend its specificity. The difference from a concept classifier is that, in the hierarchical index, a specific concept may have more than one super-ordinate concept to which it extends the meaning. For instance, the concept 保険料免除期間 *(insurance premium exemption period)* is more specific than both concepts 保険料 *(insurance premium)* in Figure 5.1a and 期間 *(period)* in Figure 5.1b.



(a) Hierarchical index of concept *Insurance premium*.



(b) Hierarchical index of concept *Period*.

Figure 5.1: Examples of hierarchical index for legal concepts from the Japanese National Pension Act.

Generating hierarchical structure of legal indices is an ontology learning problem, in which the super-ordinate and sub-ordinate relations among legal concepts are considered. Basically, the problem of ontology learning from unstructured text includes two main steps: extracting concepts and discovering the relations among the concepts. The first step is usually treated as a terminology extraction task while the second step has several ways. In general, there are two common lines of approach to discover the hierarchies of concepts [WLB12]: statistics-based ontology learning and linguistics-based ontology learning. In legal ontology learning, most of studies applied the second line of approach by which researchers discover the hierarchy of ontology for many languages (e.g. German, French, Portuguese, Thai, Tunisian) using taxonomical chains [LMPV07, LMPV09, SQ05], syntactic structure analysis [Lam05], document logical structure [MG14] or extending a seed ontology [BS12] using a lexical resource such as WordNet. As we do literature review, there is no work on Japanese legal ontology learning which is published in English. Our work serves as effort in generating Japanese legal ontology based on natural language processing approach in which only super-ordinate and sub-ordinate relations are considered.

Legal indices are terms containing the main information in legal documents. The task of extracting Japanese legal indices is treated as a keyword extraction problem and assumed to be ready with existing work [LNS13]. We propose an approach to generate the hierarchical index for legal documents by exploiting directional similarity [CM05] to determine the super/sub-ordinate relationships among the legal concepts. Since super-ordinate and sub-ordinate relations are one-way relations and they have reverse meaning in each pair of concepts, the super-ordinate relations can be inferred from sub-ordinate relations and vice versa. Therefore, we express one of them and indicate the relation between a pair of concept as a direction. For a set of legal concepts, hereby *legal indices*, the directional relations among them form a directed graph. In this graph, the cycles caused the synonyms of the legal indices are eliminated by combining the synonym indices in the same vertices. Then, the direct relation between any pair of vertices is omitted if there exists an indirect path linking that pair. The hierarchical index is generated by exporting all vertices with its child vertices recursively.

We adopt the idea of hierarchical index to a document of Japanese National Pension Act. This work serves as effort in extracting the semantic relations from Japanese legal documents by natural language processing approach. With the representation as a hierarchical structure, the users can structurally imagine the overview of the main legal concepts. The hierarchical structure is evaluated based on the number of correct relations in comparison with an annotated legal ontology. We achieved 40.6% of precision, 46.9% of recall and 43.5% of F-measure as the performance of the proposed method.

The rest of this chapter is as following: Section 5.2 overviews the approaches which are tailored for legal ontology construction; Section 5.3 introduces our proposal to construct Japanese legal ontology and extract hierarchy from this ontology; Section 5.4 describes the experiments; and Section 5.5 summarizes the contents of this chapter.

5.2 Related Work

Since the hierarchical index of legal documents represents the super-ordinate and sub-ordinate relations among legal indices, this structure is a kind of legal ontology which considers only super-ordinate and sub-ordinate relations among legal concepts. The term *ontology* originates in philosophy and is described as "a subject of study that is concerned with the nature of existence[1]." In Computer Science, an ontology is defined as an explicit specification of a conceptualization [Gru93] and often represented by formal languages such as Web Ontology Language (OWL). When applied to the legal domain, legal ontology is an explicit and formal description of the legal concepts and their relations. Researchers address several uses [BCBG05, Val05] of the legal ontologies such as: *i*) organize and structure information; *ii*) reasoning and problem solving; *iii*) semantic indexing and search; *iv*) semantic integration/interoperation; and *v*) understand a domain.

For the purpose of knowledge representation, ontology plays an important role in many domains, such as a representation to understand the general relations of the main knowledge concepts, and a knowledge base for information retrieval system. Hence, researchers are expending significant effort on methodologies for automatically constructing ontology to describe the relations of concepts. Generally, there are two common lines of approach to discover the relations among concepts when constructing ontology automatically from text [WLB12]:

- Statistic-based techniques: discover concepts' hierarchies using techniques such as clustering concepts into groups or analyzing the co-occurrences of concepts;
- Linguistics-based techniques: uncover concepts and relations by utilizing syntactic structure and dependency information or employing semantic lexicon resource (e.g. WordNet).

As the scope of this work belongs to the research field of Legal Engineering, we overview approaches that are tailored for ontology construction in the legal domain. Corcho et al. [CFLGPLC05] show how experts in the legal domain build their own legal ontologies with METHONTOLOGY and WebODE. Many works explore computational linguistic

[1]Longman Dictionary of Contemporary English. Advanced Learner's Dictionary.

techniques to construct legal ontology automatically. These methods include two main steps: extracting the legal terms and finding the relations among terms. Lenci et al. [LMPV07, LMPV09] extract terms from Italian text and find the relations among them by taxonomical chains. Walter and Pinkal [WP06, WP09] find relations of concepts in extracted definitions to improve the quality of text-based ontology learning from German court decisions. Lame [Lam05] identifies legal ontology components for French Codes by extracting legal terms and legal concepts by natural language processing methods, then blends syntactical analysis with statistical analysis to find the relations among them. Saias and Quaresma [SQ05] extract legal terms with their properties with natural language processing tools and identify relations among them by modifiers or heads in their lexical chains; this legal ontology is then merged with an existing top-level Portuguese ontology and used in a logic programming framework EVOLP+ISCO to enhance a retrieval system. Boonchom and Soonthornphisaj [BS12] improve the performance of court sentences retrieval process by a proposal of Automatic Thai Legal Ontology Building (ATOB), which automatically generates seed ontology and expands the ontology using Thai legal terminology *TLlexicon*. Mezghanni and Gargouri [MG14] tackle the logical structures of documents and Formal Concept Analysis to construct a legal ontology for Tunisian criminal law.

5.3 Generating Hierarchical Index

As introduced in Section 5.1, the idea of the hierarchical index is applied to legal domain to represent the legal concepts hierarchically. In the hierarchical index of legal documents, the concepts of indices at higher levels are more general than the concepts of indices at lower levels on meaning aspect; or, the concepts at the lower levels are more specific than those at the higher levels. We call these kinds of relations are super-ordinate and sub-ordinate relations of the legal concepts. For convenience, we refer to the super-ordinate concepts as parents and sub-ordinate concepts as children.

The overview of our approach for generating a hierarchical index of legal document contains following steps:

1. Extract the legal indices from legal documents;
2. Construct a directed graph $G(V, E)$, where V is the set of legal indices, and E is the set of directed edges indicating the sub-ordinate directions from child indices to parent indices;
3. Eliminate the cycles caused by synonyms in $G(V, E)$;

4. Eliminate the directed edge from index C to index P, with $C, P \in V$, if there exists an indirect path from index C to index P;

5. Export the hierarchical index of legal concepts from the directed graph.

The legal indices are words/phrases that express the important contents of legal documents. Extracting legal indices is treated as a task of extracting keywords from legal documents which is available with existing work by Le et al. [LNS13]. The authors assume that Japanese keywords are occurred in chunks and include important words or tokens. They assign weights to express the importance of words and chunks, then collect highly weighted chunks as the keywords for Japanese legal documents.

By observation of Japanese morphology, child indices usually expand the parent indices

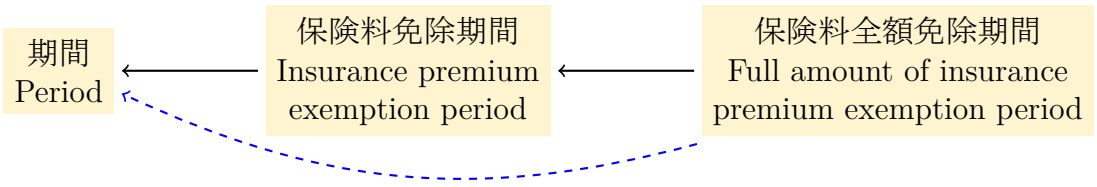


Figure 5.2: The directional relations of three legal indices. The directions of edges indicate the sub-ordinate relations from child indices to parent indices. The lines and the dashed line are respectively the explicit and the implicit relations.

by adding words. For example, the index 保険料改定率 *(insurance premium revision rate)* extends the specificity of the index 改定率 *(revision rate)* by adding word 保険料 *(insurance premium)*. Hence, the hierarchical structure can simply be constructed by counting the overlapping of taxonomical chains occurring in the indices. However, the positions of inserted words in child indices are not fixed and can occur before, after or in the middle of the parent indices. For instance, the index 保険料全額免除期間 *(full amount of insurance premium exemption period)* is more specific than the index 保険料免除期間 *(insurance premium exemption period)* due to the addition of word 全額 *(full amount)* in the middle of the lexical chain. In addition, the overlapping words in child indices do not exactly correspond with the parent indices. For instance, the index 被用者年金各法 *(employee pension acts)* has sub-ordinate relation to the index 法令 *(laws and regulations)* since the word 法 *(act)* is a sub-class of word 法令 *(laws and regulations)*. Therefore, we consider the overlapping of words on the semantic aspects and the relations of the indices are determined by their meaning.

Since a super-ordinate relation from a parent index to a child index can be inferred from the sub-ordinate relation from the child index to the parent index and vice versa, only one kind of relations is adequate to express the relation between two indices. We

choose one, i.e. the sub-ordinate relation, and indicate this relation by a direction. When connecting all legal indices together, the sub-ordinate relations structure a directed graph. Therefore, to represent the relations among the legal indices, we employ a directed graph in which the directions of edges express the sub-ordinate relations of indices. An advantage of representing the relations among indices by directions is that the subsumption of grant parent indices to grant child indices is also revealed. Figure 5.2 illustrates the relations of three legal indices where the directions of edges indicate the sub-ordinate relations from child indices to parent indices. In this example, the directions indicate that the index 保険料全額免除期間 *(full amount of insurance premium exemption period)* is more specific than the index 保険料免除期間 *(insurance premium exemption period)* and the index 保険料免除期間 *(insurance premium exemption period)* is more specific than the index 期間 *(period)*. Without having a directed edge, we still understand the implicit sub-ordinate relation from the index 保険料全額免除期間 *(full amount of insurance premium exemption period)* to the index 期間 *(period)*.

5.3.1 Constructing Directed Graph of Legal Indices

As analyzed in previous section, the directed graph is employed to represent the relations among legal indices. In this directed graph, vertices express the legal indices and edges express the directions of sub-ordinate relations. We determine the directions among the legal indices using directional similarity. The semantic similarity (or semantic relatedness) is a metric indicating the distance of meaning between two concepts or texts. Directional similarity [CM05] is defined as the semantic similarity of a text segment T_i with respect to another text segment T_j. Specifically, directional similarity measures how similar of a text, T_i, is to another text, T_j. The directional similarity of T_i with respect to T_j is based on the semantic similarities of words including in both text segments. In turn, the semantic similarity of words is treated differently based the types: semantic similarities of nouns and verbs are based on their distances in a thesaurus such as WordNet while the similarities of adjectives, adverbs, cardinal numbers are based on lexical matching.

Since the legal indices are not very long as a paragraph and include the important words, we treat all words in legal indices equally and measure the word-to-word similarities based on WordNet using metrics defined on the concept distances such as Lin [Lin98] and WuPalmer [WP94]. Then, the directional similarity of an index C with respect to an index P is modified as follows:

$$Sim(C,P) = \frac{\sum_{w_i \in C} maxSim(w_i) * idf_{w_i}}{\sum_{w_i \in C} idf_{w_i}} \tag{5.1}$$

in which, $maxSim(w_i)$ is the highest word-to-word semantic similarity of word w_i to words in P and idf_{w_i} is the Inverse Document Frequency (IDF) score of the word w_i. The word-to-word semantic similarity between two words w_i and w_j is measured based on the distance of their concepts [WP94, Lin98] in a semantic network such as WordNet [Mil95, Fel98]. Note that, in this study, we treat the synonyms as the same words and word-to-word similarity of a word and its synonyms is 1. The IDF score of a word w indicates the importance of the word w in a collection of N documents and is calculated as:

$$idf_w = \log \frac{N}{df_w} \tag{5.2}$$

where df_w is the document frequency or the number of documents in which the word w appears.

The directional similarity score has value from 0 to 1 indicating the degree of similarity of an index to another index. When the directional similarity of index P to index C is $Sim(P, C) = 1$, it means that index P is exactly the same with index C in the meaning aspect while it does not mean that index C is also exactly the same with index P. This fact is caused by the extension of P's specificity in concept C when adding new word(s). Therefore, we determine the sub-ordinate relation from C to P based on their directional similarity. In brief, we determine the sub-ordinate relation of index C to index P if they occur in the same articles and the directional similarity of index C with respect to index P is $Sim(P, C) = 1$.

Intuitively, we take two examples for two pairs of legal indices:

- C_1 = 保険料全額免除期間 *(full amount of insurance premium exemption period)* and P_1 = 保険料免除期間 *(insurance premium exemption period)* refer to the period of insurance exemption, in which C_1 is more specific than P_1;
- C_2 = 被用者年金各法 *(employee pension acts)* and P_2 = 法令 *(laws and regulations)* refer to the law, in which C_2 is more specific than P_2.

Given the IDF scores of all words shown in Table 5.1 and the word-to-word semantic similarities by WuPalmer metric of all pairs of words in two pairs of indices as in Table 5.2 and Table 5.3, the directional similarity of each pair of indices is:

- Directional similarity of C_1 to P_1 is $Sim(C_1, P_1) = 0.63$;
- Directional similarity of P_1 to C_1 is $Sim(P_1, C_1) = 1.0$;
- Directional similarity of C_2 to P_2 is $Sim(C_2, P_2) = 0.23$;
- Directional similarity of P_2 to C_2 is $Sim(P_2, C_2) = 1.0$.

When constructing the directed graph of the legal indices, the directions between pairs of indices indicate the sub-ordinate relations or the directions forward from specific concepts to general concepts. Assume that each pair of indices occur in the same articles, the direction is determined to forward from index C to index P if the directional similarity of P to C is $Sim(P, C) = 1$. For the two pairs of indices in the example, the directed edges forward from index C_1 to index P_1 and from index C_2 to index P_2 to indicate that indices C_1 and C_2 are sub-ordinate indices of P_1 and P_2 respectively. Figure 5.3 shows a piece of directed graph of 11 legal indices where the directions explicit the sub-ordinate relations from the child indices to their parent indices.

Table 5.1: The IDF scores of words (tokens) in the indices 保険料全額免除期間 *(full amount of insurance premium exemption period)*, 保険料免除期間 *(insurance premium exemption period)*, 被用者年金各法 *(employee pension acts)* and 法令 *(laws and regulations)*.

Word w	**Translation**	idf_w	**Word** w	**Translation**	idf_w
保険	insurance	0.61	被用者	employee	1.76
料	premium/fee	0.31	年金	pension	0.81
全額	full amount	1.29	各	each/all	0.16
免除	exemption	1.02	法	act	0.01
期間	period	0.29	法令	laws (and regulations)	0.50

Table 5.2: The semantic similarities of all pairs of words in two legal indices 保険料全額免除期間 *(full amount of insurance premium exemption period)* and 保険料免除期間 *(insurance premium exemption period)*.

	保険 (insurance)	料 (premium/fee)	全額 (full amount)	免除 (exemption)	期間 (period)
保険 (insurance)	1.00	0.59	0.00	0.35	0.38
料 (premium/fee)	0.59	1.00	0.00	0.38	0.40
免除 (exemption)	0.35	0.38	0.00	1.00	0.53
期間 (period)	0.38	0.40	0.00	0.53	1.00

Table 5.3: The semantic similarities of all pairs of words in two legal indices 被用者年金各法 *(employee pension acts)* and 法令 *(laws and regulations)*.

	被用者 (employee)	年金 (pension)	各 (each/all)	法 (act)
法令 (laws and regulations)	0.21	0.30	0.00	1.00

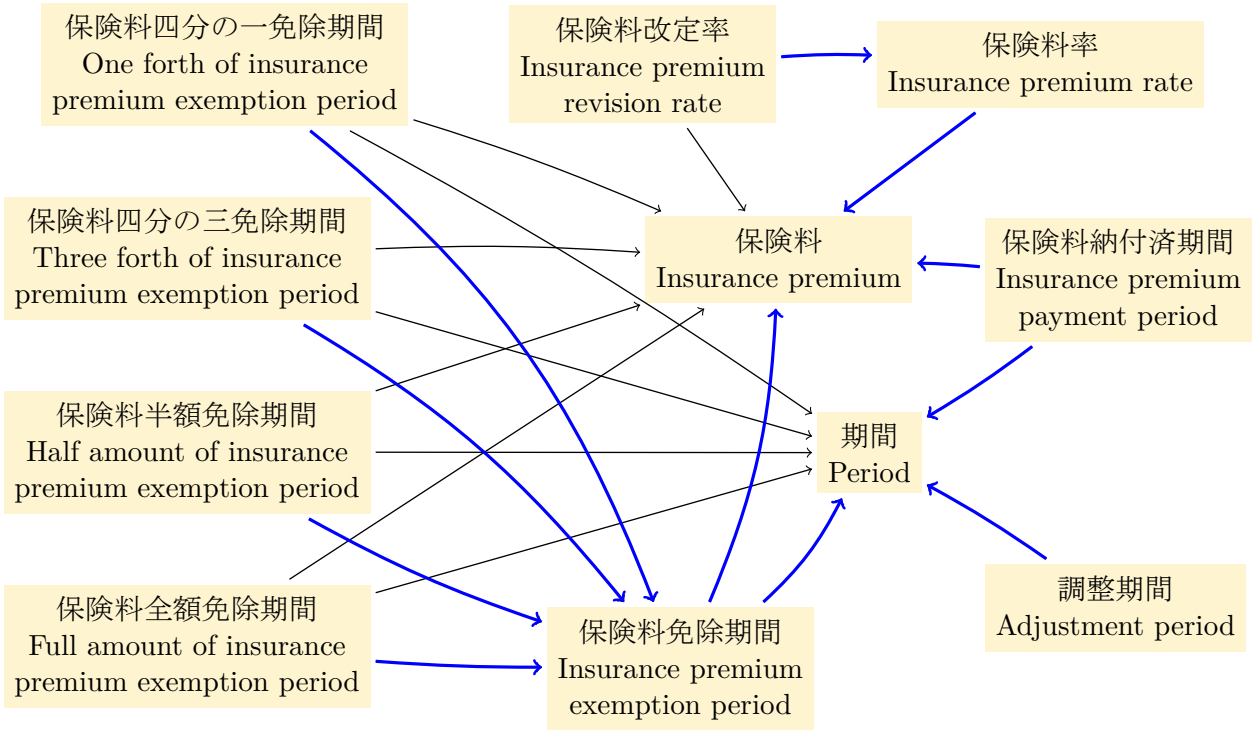


Figure 5.3: The sub-ordinate relations among a set of Japanese legal indices are represented by the arrows. The thin edges are the redundant relations which will be eliminated. The thick edges are the remaining relations after the elimination.

5.3.2 Eliminating Cycles of Synonyms

Some concepts are synonyms in that they have bi-directional relations between pairs of vertices. For example, since two indices $C =$ 金額 *(amount of money)* and $P =$ 額 *(amount)* imply the same meaning, the directional similarity with respect to both indices is the same $Sim(C, P) = Sim(P, C) = 1$. Hence, there are a direct edge from P to C and a direct edge from C to P. In the case of such synonym, the synonym indices are combined as one vertex in the directed graph with reserving their relations to and from other vertices.

5.3.3 Eliminating Unnecessary Directed Edges

Every vertex of specific legal concepts has directions to all of its general concepts. Since the sub-ordinate relations can be understand implicitly, the directed edge between two vertices is not necessary if there is another possible path connecting them. For instance, as shown in Figure 5.3, there are two paths from the index 保険料全額免除期間 *(full amount of insurance premium exemption period)* to the index 期間 *(period)*: a direct path and an indirect path via the index 保険料免除期間 *(insurance premium exemption period)*. As analyzing previously in Figure 5.2, we are still able to understand that the index 保険料全額免除期間 *(full amount of insurance premium exemption period)* has a sub-ordinate relation to the index 期間 *(period)* without an explicit sub-ordinate relation but via the index 保険料免除期間 *(insurance premium exemption period)*. Hence, the directed edge between them is redundant. Therefore, the directed edge between any two vertices is eliminated if there exists an indirect path connecting two vertices. In Figure 5.3, the bold edges indicate the sub-ordinate relations among legal indices after eliminating unnecessary directed edges.

5.3.4 Exporting Hierarchical Index

When having the directed graph of legal indices without synonyms and unnecessary edges, all indices in the directed graph are exported as a table of hierarchical index for legal documents as follows:

1. Sort vertices by ascending order of outdegree. Note that, the outdegree of a vertex in a directed graph is defined as the number of outward edges from that vertex.

2. For each vertex, print its index and its child indices from incoming edges recursively. Vertices are marked as visited when its index has been printed to the hierarchy.

3. Print the rest of the indices which have not been visited.

5.4 Experiments

We adopt the idea of the hierarchical index on Japanese National Pension Act (JNPA) document which includes 208 indices. Cabocha[2] [KM02] is employed as parser to separate the words in the legal indices. Japanese WordNet[3] [BIF+09] is used as the thesaurus to

[2]Cabocha is available at `https://code.google.com/p/cabocha/`.
[3]The Japanese WordNet is available at `http://nlpwww.nict.go.jp/wn-ja/index.en.html`.

calculate the semantic similarity of words. We use JAWJAW[4] as tool to compute the word-to-word similarity based on WordNet thesaurus. Two metrics for word-to-word semantic similarity are WuPalmer [WP94] and Lin [Lin98]. The corpus to calculate IDF scores for words includes 7,984 legal documents from the Japanese government web page updated until July 31st, 2013.

The structure of the generated hierarchy is evaluated using an annotated ontology made for JNPA by specialists. This ontology describes the IS-A relations of the legal indices which may not be included in the JNPA document. Therefore, we extract a subset of indices which are included in the JNPA document with their corresponding relations as the benchmark for evaluation. The annotated hierarchical index contains 208 indices with 115 sub-ordinate relations. We evaluate the structure of generated hierarchy by counting the number of correct sub-ordinate relations. Since there are many indices which do not have sub-ordinate relations to more general concepts, we consider them to have sub-ordinate relations to NULL. In addition, the synonyms are separated and count all sub-ordinate relations among them. The relations to and from the synonym indices are also increased by the separation. Finally, in the annotated hierarchy, there are 226 sub-ordinate relations among 208 indices of the JNPA document.

We use Precision, Recall and F-measure as metrics to evaluate the overlap of the generated and annotated structures. The Precision is the percentage of correct sub-ordinate relations on the generated relations:

$$Precision = \frac{\text{\# Correct Sub-ordinate Relations}}{\text{\# Generated Sub-ordinate Relations}}$$

The Recall is the percentage of the correct relations on the annotated relations:

$$Recall = \frac{\text{\# Correct Sub-ordinate Relations}}{\text{\# Annotated Sub-ordinate Relations}}$$

F-measure is the harmonic score for Precision and Recall:

$$F\text{-}measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The hierarchical index for the JNPA document has been generated using the method described previously and the extracted relations has been count as the same fashion with the annotated data. The performance of our proposed approach is shown in Table 5.4. Note that, *# Extract* is the number of extracted sub-ordinate relations and *# Correct* is

[4]JAWJAW is a Java API for Japanese WordNet-based semantic similarity which is available at `http://www.cs.cmu.edu/~hideki/software/jawjaw/index-en.html`.

Table 5.4: The evaluation on the structure of the generated hierarchical index for JNPA document.

Metrics	**# Extract**	**# Correct**	**Prec.**	**Rec.**	**F-measure**
WuPalmer	266	106	39.8%	46.9%	43.1%
Lin	261	106	40.6%	46.9%	43.5%

the number of correct sub-ordinate relations. For both similarity metrics, we extracted 106 correct relations, roughly a half of the annotated relations. The overall performances of our approach using two metrics are almost the same, more than 43% on F-measure. In this experiment, the hierarchy resulted with Lin metric achieves slightly higher precision. Hence, we obtained better performance with Lin metric at 40.6% for precision, 46.9% for recall and 43.5% for F-measure.

Error Analysis

We analyze the generating performance on the hierarchical index for the JNPA document by Lin metric: 155 sub-ordinate relations are wrongly extracted and 120 annotated relations are missed. The wrongly extracted relations are caused by:

- The overlapping of tokens, such as token 障害 *(disability)* features a sub-ordinate relation from index 障害基礎年金 *(disability basic pension)* to index 障害 *(disability)*;
- The overlapping of the synonym tokens, such as tokens 料 *(fee/premium)* and 費用 *(cost)* are recognized as synonyms in the Japanese WordNet, therefore an sub-ordinate relation is formulated from index 保険料 *(insurance premium)* to index 費用 *(cost)*; and
- Other reasons, such as the co-occurrence constrains prevent index 返還金債権 *(claim for refund)* to be considered to have relation with index 権利 *(right)* since they do not occur in the same article.

The annotated relations are missed because the relations between pairs of indices are not featured due to the similarities. For instance, index 老齢基礎年金 *(old age basic pension)* is annotated as a descendant of index 年金給付 *(pension benefit)*, however, the directional similarity does not recognize the relation between them.

5.5 Conclusions

Due to the large number of legal documents and the complexity of the legal information, the readers may have difficulties in reading and understanding the contents. We aim to support them by providing a briefly structured representation of the legal concepts. The idea of hierarchical index is applied to the legal domain to represent the super/sub-ordinate relations among Japanese legal concepts. This work serves as effort in the task of Japanese legal ontology learning by language processing approach, in which the super-ordinate and sub-ordinate relations among legal concepts are considered. First, legal indices that reveal the main contents of legal documents are extracted. Second, relations among legal indices are indicated by directions and determined based on directional similarity. Then, relations among a set of legal indices structure a directed graph. Third, the cycles caused by synonyms and unnecessary relations in the directed graph are eliminated. Finally, this directed graph is exported as a table of hierarchical structure of legal indices. This idea has been experimented on the Japanese National Pension Act document and the generated hierarchical structure has been evaluated based on an annotation. We achieved 43.5% of F-measure as the overall performance of the proposed method. In the future, we plan to extend this work by exploring the semantic role labels and logical structure of legal documents on the indices to discover the relationships among legal indices.

Chapter 6

Conclusion Remarks and Future Work

6.1 Conclusions

For the reasons of the information load and the complexity of the contents in the system of legal documents, both the specialists and non-specialists have difficulties in searching and understanding the legal information. In this research, we introduce a hierarchical structure of legal indices to provide the readers a general view on the relations among the legal concepts. We divide the problem of constructing the hierarchy into two tasks: extracting important concepts and discovering the relations among these concepts. The first task, extracting important concepts, is treated as a keyphrase extraction problem. The second task, discovering the relationships among concepts, is treated as a problem of legal ontology construction. The main contributions of this dissertation are summarized as follows:

1. We addressed the problem of automatically extracting legal indices which express the important contents of legal documents. Legal indices are not limited to single-word keywords and compound-word (or phrase) keywords, they are also clause keywords. We approach index extraction using structural information of Japanese sentences, i.e. chunks and clauses. Based on the assumption that legal indices are composed of important tokens from the documents, extracting legal indices is treated as a problem of collecting chunks and clauses that contain as many important tokens as possible. Each token is assigned a weight which are statistical scores, e.g. TF-IDF and Okapi BM25, to indicate its importance. The importance of a chunk or a clause is determined based on the average weights of tokens included in that chunk or clause. Then, highly weighted chunks and clauses are recognized as the indices for

legal documents. The experimental results on Japanese National Pension Act data show that our proposed method achieves better performance (8.6% higher on F1-score) than TextRank, the most popular unsupervised method in extracting single-word and compound-word keywords. In addition, this proposal is also applicable to extract clause keywords with high performance.

2. We improved the performance of English keyphrase extraction and involved new kinds of words to English keyphrases. When analyzing the English keyphrase extraction approaches, we realize that current studies often extract keyphrases by collecting adjacent important adjectives and nouns. However, the statistics on four public corpora shows that about 15% of keyphrases contain other kinds of words such as present/past participles, comparative/superlative adjectives and cardinal numbers. Even so, incorporating such kinds of words to the noun phrase patterns is not a solution to improve the extraction performance. Therefore, we propose a solution to improve the extraction performance by involving new kinds of words to keyphrases. First, keyphrase candidates are extracted from noun phrases using syntactic information which is obtained by shallow and deep parsing. Second, candidates are then associated with weights to indicate their importance in documents. The weight of a noun phrase candidate is computed as the average of the weights of tokens in it. Finally, the top weighted candidates in each document are selected as keyphrases for that document. We have experimented on four public corpora to demonstrate that our proposal improve the performance of keyphrase extraction and new kinds of words are introduced to keyphrases. In addition, our proposal is also superior to the current unsupervised keyphrase extraction approaches.

3. We applied the idea of hierarchical index the legal domain to provide the readers a general understanding of legal concepts via their super/sub-ordinate relations. This work serves as effort in automatic legal ontology learning in which super/sub-ordinate relations are considered. Indices are extracted from legal documents as keywords and their relationships are discovered by language processing method. We propose an approach to extract the super/sub-ordinate relation between each pair of concepts individually based on directional similarity. The relations among a set of legal indices are represented in a directed graph and the hierarchical structure of indices is simply exported from this graph. We adopt this proposal to the Japanese National Pension Act document. The resulted hierarchical structure is compared to an annotated legal ontology on the number of correct relations. The proposed method achieves 40.6% for precision, 46.9% for recall and 43.5% for F-measure as the performance.

6.2 Future Research Directions

In the future, we are going to improve the limitations and extend this dissertation in the following directions:

1. In chapter 3, we presented the extraction of Japanese legal keyphrases. In the future, we are going to apply the proposed approach to other domains such as news or scientific papers to examine the performance of Japanese keyphrase extraction approach.

2. In chapter 4, we introduced new kinds of words to English keyphrases and improved the extraction performance. The proposed approach employs only statistical scores of individual words and the parsing information of sentences. We intend to improve the performance by adding the context information, such as looking at the relatedness of the individual words to the topic of the documents to enhance the set of candidates.

3. In chapter 5, we introduced the hierarchical structure of legal indices and proposed a language processing based approach to discover the relations among legal indices. In the future, we plan to extend this work by exploring the semantic role labels and logical structure of legal documents on the indices to discover the relationships among legal indices.

4. For the general purpose, we intend to extend this study by applying the generated ontology to a retrieval system. Then, we will be able to evaluate how much the knowledge-base benefit legal information retrieval for Japanese legal documents.

Bibliography

[AB03] Kevin D. Ashley and Stefanie Brüninghaus. A predictive role for intermediate legal concepts. In *Legal Knowledge and Information Systems. Jurix 2003: The Sixteenth Annual Conference*, pages 1–10, 2003.

[Aiz03] Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing and Management: an International Journal*, 39(1):45–65, 2003.

[BA99] Stefanie Brüninghaus and Kevin D. Ashley. Toward adding knowledge to learning algorithms for indexing legal cases. In *Proceedings of the 7th International Conference on Artificial Intelligence and Law*, ICAIL '99, pages 9–17, 1999.

[BBD13] Adrien Bougouin, Florian Boudin, and Béatrice Daille. Topicrank: Graph-based topic ranking for keyphrase extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 543–551, 2013.

[BCBG05] V. Richard Benjamins, Pompeu Casanovas, Joost Breuker, and Aldo Gangemi. Law and the semantic web, an introduction. In V. Richard Benjamins, Pompeu Casanovas, Joost Breuker, and Aldo Gangemi, editors, *Law and the Semantic Web*, pages 1–17. Springer-Verlag, 2005.

[BIF+09] Francis Bond, Hitoshi Isahara, Sanae Fujita, Kiyotaka Uchimoto, Takayuki Kuribayashi, and Kyoko Kanzaki. Enhancing the japanese wordnet. In *Proceedings of the 7th Workshop on Asian Language Resources*, ALR7, pages 1–8, 2009.

[BJL+99] Douglas Biber, Stig Johansson, Geoffrey Leech, Susan Conrad, and Edward Finegan. *Longman grammar of spoken and written English*. Longman, 1999.

[BM85] David C. Blair and M. E. Maron. An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of the ACM*, 28(3):289–299, 1985.

[BS12] Vi-sit Boonchom and Nuanwan Soonthornphisaj. Atob algorithm: an automatic ontology construction for thai legal sentences retrieval. *Journal of Information Science*, 38(1):37–51, 2012.

[CBGG14] Cornelia Caragea, Florin Adrian Bulgarov, Andreea Godea, and Sujatha Das Gollapalli. Citation-enhanced keyphrase extraction from research papers: A supervised approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, EMNLP '14, pages 1435–1446, 2014.

[CFLGPLC05] Oscar Corcho, Mariano Fernández-López, Asunción Gómez-Pérez, and Angel López-Cima. Building legal ontologies with methontology and webode. In V. Richard Benjamins, Pompeu Casanovas, Joost Breuker, and Aldo Gangemi, editors, *Law and the Semantic Web*, pages 142–157. Springer-Verlag, 2005.

[CM05] Courtney Corley and Rada Mihalcea. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, EMSEE '05, pages 13–18, 2005.

[Fel98] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[FPW+99] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. Domain-specific keyphrase extraction. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 668–673, 1999.

[GA11] Matthias Grabmair and Kevin D. Ashley. Facilitating case comparison using value judgments and intermediate legal concepts. In *Proceedings of the 13th International Conference on Artificial Intelligence and Law*, ICAIL '11, pages 161–170, 2011.

[GM07] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 1606–1611, 2007.

[Gru93] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition - Special issue: Current issues in knowledge modeling*, 5(2):199–220, 1993.

[HN10] Kazi Saidul Hasan and Vincent Ng. Conundrums in unsupervised keyphrase extraction: Making sense of the state-of-the-art. In *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume*, COLING '10, pages 365–373, 2010.

[HN14] Kazi Saidul Hasan and Vincent Ng. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '14, pages 1262–1273, 2014.

[Hul03] Anette Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 216–223, 2003.

[JC97] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc of 10th International Conference on Research in Computational Linguistics, ROCLING' 97*, 1997.

[Jon72] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[Kat07] Takuya Katayama. Legal engineering - an engineering approach to laws in e-society age. In *Proceedings of the First International Workshop JURISIN*, pages 1–5, 2007.

[KM02] Taku Kudo and Yuji Matsumoto. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, pages 1–7, 2002.

[KMKB10] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 21–26, 2010.

[Lam05] Guiraude Lame. Using nlp techniques to identify legal ontology components: Concepts and relations. In V. Richard Benjamins, Pompeu

Casanovas, Joost Breuker, and Aldo Gangemi, editors, *Law and the Semantic Web*, pages 169—184. Springer-Verlag, 2005.

[LC98] C. Leacock and M. Chodorow. *Combining local context and WordNet sense similiarity for word sense disambiguation*, pages 265–283. The MIT Press, 1998.

[LD97] Thomas K Landauer and Susan T. Dutnais. A solution to plato' s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

[Les86] Michael E Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, SIGDOC '86, pages 24–26, 1986.

[LHZS10] Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 366–376, 2010.

[Lin98] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, 1998.

[LL08] Marina Litvak and Mark Last. Graph-based keyword extraction for single-document summarization. In *Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization*, MMIES '08, pages 17–24, 2008.

[LLZS09] Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 257–266, 2009.

[LMPV07] Alessandro Lenci, Simonetta Montemagni, Vito Pirrelli, and Giulia Venturi. Nlp-based ontology learning from legal texts. a case study. In *Proceedings LOAIT07-II Workshop on Legal Ontologies and Artificial Intelligence Technique*, pages 113—129, 2007.

[LMPV09] Alessandro Lenci, Simonetta Montemagni, Vito Pirrelli, and Giulia Venturi. Ontology learning from italian legal texts. In *Proceedings of the 2009 Conference on Law, Ontologies and the Semantic Web: Channelling the Legal Information Flood*, pages 75–94, 2009.

[LNS13] Tho Thi Ngoc Le, Minh Le Nguyen, and Akira Shimazu. Unsupervised keyword extraction for japanese legal documents. In *Proceedings of Legal Knowledge and Information Systems - JURIX 2013: The Twenty-Sixth Annual Conference*, pages 97–106, 2013.

[LPLL09] Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 620–628, 2009.

[LR10] Patrice Lopez and Laurent Romary. Humb: Automatic key term extraction from scientific articles in grobid. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 248–251, 2010.

[MA03] Marie-Francine Moens and Roxana Angheluta. Concept extraction from legal cases: The use of a statistic of coincidence. In *Proceedings of the 9th International Conference on Artificial Intelligence and Law*, ICAIL '03, pages 142–146, 2003.

[Mat99] Jérôme Mathieu. Adaptation of a keyphrase extractor for japanese text. In *Proceedings of the 27th Annual Conference of the Canadian Association for Information Science*, CAIS '99, pages 182–189, 1999.

[MBK00] Charles T. Meadow, Bert R. Boyce, and Donald H. Kraft. *Text Information Retrieval Systems*. Academic Express Inc., 2nd edition, 2000.

[McB10] Nicholas J. McBride. *Letters to a Law Student: A Guide to Studying Law at University*. Pearson Longman, 2nd edition, 2010.

[MG14] Imen Bouaziz Mezghanni and Faiez Gargouri. Learning of legal ontology supporting the user queries satisfaction. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 01, pages 414–418, 2014.

[MI03] Yutaka Matsuo and Mitsuru Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. In *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference*, pages 392–396, 2003.

[MI04] Yutaka Matsuo and Mitsuru Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13:157–169, 2004.

[Mil95] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[MKKT04] Takehiko Maruyama, Hideki Kashioka, Tadashi Kumano, and Hideki Tanaka. Development and evaluation of japanese clause boundaries annotation program. *Natural Language Processing*, 11(3):39—68, 2004. (In Japanese).

[MS99] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.

[MS08] K. Tamsin Maxwell and Burkhard Schafer. Concept and context in legal information retrieval. In *Proceedings of the 2008 Conference on Legal Knowledge and Information Systems: JURIX 2008: The Twenty-First Annual Conference*, pages 63–72, 2008.

[MT04] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, EMNLP '04, pages 404–411, 2004.

[Neg14] Sumit Negi. Single document keyphrase extraction using label information. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1468–1476, 2014.

[NK07] Thuy Dung Nguyen and Min-Yen Kan. Keyphrase extraction in scientific publications. In *Proceedings of International Conference on Asian Digital Libraries*, pages 317–326, 2007.

[NM02] Hiroshi Nakagawa and Tatsunori Mori. A simple but powerful automatic term extraction method. In *COLING-02 on COMPUTERM 2002: Second International Workshop on Computational Terminology - Volume 14*, COMPUTERM '02, pages 1–7, 2002.

[NM03] Hiroshi Nakagawa and Tatsunori Mori. Automatic term recognition based on statistics of compound nouns and their components. *Terminology*, 9(2):201—219, 2003.

[NYM03] Hiroshi Nakagawa, Hiroaki Yumoto, and Tatsunori Mori. Term extraction based on occurrence and concatenation frequency. *Journal of Natural Language Processing*, 10(1):27–45, 2003. (In Japanese).

[OM97] Yasushi Ogawa and Toru Matsuda. Overlapping statistical word indexing: A new indexing method for japanese text. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '97, pages 226–234, 1997.

[OSdCI11] Jesús Oliva, José Ignacio Serrano, María Dolores del Castillo, and Ángel Iglesias. Symss: A syntax-based measure for short-text semantic similarity. *Data & Knowledge Engineering*, 70(4):390–405, 2011.

[Ove01] Paul Over. Introduction to duc-2001: an intrinsic evaluation of generic news text summarization systems. In *Proceedings of the International 2001 Document Understanding Conference*, 2001.

[PBP03] Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'03, pages 241–257, 2003.

[PR00] Vasin Punyakanok and Dan Roth. The use of classifiers in sequential inference. In *Proceedings of Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS)*, pages 995–1001, 2000.

[Res95] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, IJCAI'95, pages 448–453, 1995.

[RU11] Anand Rajaraman and Jeffrey David Ullman. *Mining of Massive Datasets*, chapter Data Mining, pages 1–17. Cambridge University Press, New York, NY, USA, 2011.

[RWJ+94] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference*, TREC '94, pages 109–126, 1994.

[Sal71] Gerard Salton. *The SMART Retrieval System-Experiments in Automatic Document Processing*. Prentice-Hall, Inc., 1971.

[SB88] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journa*, 24(5):513–523, 1988.

[SFS97] Yoshimi Suzuki, Fumiyo Fukumoto, and Yoshihiro Sekiguchi. Keyword extraction of radio news using term weighting for speech recognition. In *Proceedings of Natural Language Processing Pacific Rim Symposium 1997*, NLPRS'97, pages 301–306, 1997.

[SFS98] Yoshimi Suzuki, Fumiyo Fukumoto, and Yoshihiro Sekiguchi. Keyword extraction using term-domain interdependence for dictation of radio news. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, pages 1272–1276, 1998.

[SL68] G. Salton and M. E. Lesk. Computer evaluation of indexing and text processing. *Journal of the ACM*, 15(1):8–36, 1968.

[SQ05] José Saias and Paulo Quaresma. A methodology to create legal ontologies in a logic programming information retrieval system. In V. Richard Benjamins, Pompeu Casanovas, Joost Breuker, and Aldo Gangemi, editors, *Law and the Semantic Web*, pages 185–200. Springer-Verlag, 2005.

[SRR09] M. Saravanan, B. Ravindran, and S. Raman. Improving legal information retrieval using an ontological framework. *Artificial Intelligence and Law*, 17(2):101–124, 2009.

[Tur99] Peter D. Turney. Learning to extract keyphrases from text. Technical Report ERB-1057, National Research Council Canada, Institute for Information Technology, 1999.

[Tur00] Peter D. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, 2000.

[Tur01] Peter D. Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the 12th European Conference on Machine Learning*, EMCL '01, pages 491–502, 2001.

[TVV10] George Tsatsaronis, Iraklis Varlamis, and Michalis Vazirgiannis. Text relatedness based on a word thesaurus. *Journal of Artificial Intelligence Research*, 37(1):1–40, 2010.

[Val05] Andre Valente. Types and roles of legal ontologies. In V. Richard Benjamins, Pompeu Casanovas, Joost Breuker, and Aldo Gangemi, editors, *Law and the Semantic Web*, pages 65–76. Springer-Verlag, 2005.

[WLB12] Wilson Wong, Wei Liu, and Mohammed Bennamoun. Ontology learning from text: A look back and into the future. *ACM Computing Surveys*, 44(4):20:1–20:36, 2012.

[WP94] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, 1994.

[WP06] Stephan Walter and Manfred Pinkal. Automatic extraction of definitions from german court decisions. In *Proceedings of the Workshop on Information Extraction Beyond The Document*, pages 20–28, 2006.

[WP09] Stephan Walter and Manfred Pinkal. Definitions in court decisions –automatic extraction and ontology acquisition. In *Proceedings of the 2009 Conference on Law, Ontologies and the Semantic Web: Channelling the Legal Information Flood*, pages 95–113, 2009.

[WX08a] Xiaojun Wan and Jianguo Xiao. Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 969–976, 2008.

[WX08b] Xiaojun Wan and Jianguo Xiao. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 855–860, 2008.

[WZO10] Wei Wu, Bin Zhang, and Mari Ostendorf. Automatic generation of personalized annotation tags for twitter users. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 689–692, 2010.

[YN05] Minoru Yoshida and Hiroshi Nakagawa. Automatic term extraction based on perplexity of compound words. In *Proceedings of the Second International Joint Conference on Natural Language Processing*, IJCNLP'05, pages 269–279, 2005.

[ZJH+11] Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achananuparp, Ee-Peng Lim, and Xiaoming Li. Topical keyphrase extraction from twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 379–388, 2011.

[ZWZ09] Lu Zhiqiang, Shao Werimin, and Yu Zhenhua. Measuring semantic similarity between words using wikipedia. In *Proceedings of the 2009 International Conference on Web Information Systems and Mining*, WISM '09, pages 251–255, 2009.

Appendix A

Japanese Stopwords

We list 44 Japanese stopwords whose weights are reset to zero when extracting keyphrases from Japanese legal indices.

これ (this)
それ (that)
あれ (that)
この (this)
その (that)
あの (that)
ここ (here)
そこ (there)
あそこ (there)
こちら (here)
どこ (where)
彼女 (she)
彼 (he)
です (is)
あります (have)
おります (have)
います (have)
は (grammar: (particle))
が (grammar: (particle))
の (of)
に (at)
を ((prep))
だれ (who)
なに (what)
なん (what)
何 (what)
私 (I)
貴方 (you)
貴方方 (you)
我々(we)
私達 (we)
あの人 (that person)
あのかた (that person)
で (by)
え (to)
から (from)
まで (to)
より (than)
も ((prep))
どの (which)
と (and/with)
し (and)
それで (so)
しかし (but)

Appendix B

English Stopwords

We list 571 English stopwords by Salton [Sal71] which are used in extracting English keyphrases as following:

a	a's	able	about	above	according
accordingly	across	actually	after	afterwards	again
against	ain't	all	allow	allows	almost
alone	along	already	also	although	always
am	among	amongst	an	and	another
any	anybody	anyhow	anyone	anything	anyway
anyways	anywhere	apart	appear	appreciate	appropriate
are	aren't	around	as	aside	ask
asking	associated	at	available	away	awfully
b	be	became	because	become	becomes
becoming	been	before	beforehand	behind	being
believe	below	beside	besides	best	better
between	beyond	both	brief	but	by
c	c'mon	c's	came	can	can't
cannot	cant	cause	causes	certain	certainly
changes	clearly	co	com	come	comes
concerning	consequently	consider	considering	contain	containing
contains	corresponding	could	couldn't	course	currently
d	definitely	described	despite	did	didn't
different	do	does	doesn't	doing	don't
done	down	downwards	during	e	each
edu	eg	eight	either	else	elsewhere
enough	entirely	especially	et	etc	even

ever	every	everybody	everyone	everything	everywhere
ex	exactly	example	except	f	far
few	fifth	first	five	followed	following
follows	for	former	formerly	forth	four
from	further	furthermore	g	get	gets
getting	given	gives	go	goes	going
gone	got	gotten	greetings	h	had
hadn't	happens	hardly	has	hasn't	have
haven't	having	he	he's	hello	help
hence	her	here	here's	hereafter	hereby
herein	hereupon	hers	herself	hi	him
himself	his	hither	hopefully	how	howbeit
however	i	i'd	i'll	i'm	i've
ie	if	ignored	immediate	in	inasmuch
inc	indeed	indicate	indicated	indicates	inner
insofar	instead	into	inward	is	isn't
it	it'd	it'll	it's	its	itself
j	just	k	keep	keeps	kept
know	knows	known	l	last	lately
later	latter	latterly	least	less	lest
let	let's	like	liked	likely	little
look	looking	looks	ltd	m	mainly
many	may	maybe	me	mean	meanwhile
merely	might	more	moreover	most	mostly
much	must	my	myself	n	name
namely	nd	near	nearly	necessary	need
needs	neither	never	nevertheless	new	next
nine	no	nobody	non	none	noone
nor	normally	not	nothing	novel	now
nowhere	o	obviously	of	off	often
oh	ok	okay	old	on	once
one	ones	only	onto	or	other
others	otherwise	ought	our	ours	ourselves
out	outside	over	overall	own	p
particular	particularly	per	perhaps	placed	please
plus	possible	presumably	probably	provides	q
que	quite	qv	r	rather	rd

re	really	reasonably	regarding	regardless	regards
relatively	respectively	right	s	said	same
saw	say	saying	says	second	secondly
see	seeing	seem	seemed	seeming	seems
seen	self	selves	sensible	sent	serious
seriously	seven	several	shall	she	should
shouldn't	since	six	so	some	somebody
somehow	someone	something	sometime	sometimes	somewhat
somewhere	soon	sorry	specified	specify	specifying
still	sub	such	sup	sure	t
t's	take	taken	tell	tends	th
than	thank	thanks	thanx	that	that's
thats	the	their	theirs	them	themselves
then	thence	there	there's	thereafter	thereby
therefore	therein	theres	thereupon	these	they
they'd	they'll	they're	they've	think	third
this	thorough	thoroughly	those	though	three
through	throughout	thru	thus	to	together
too	took	toward	towards	tried	tries
truly	try	trying	twice	two	u
un	under	unfortunately	unless	unlikely	until
unto	up	upon	us	use	used
useful	uses	using	usually	uucp	v
value	various	very	via	viz	vs
w	want	wants	was	wasn't	way
we	we'd	we'll	we're	we've	welcome
well	went	were	weren't	what	what's
whatever	when	whence	whenever	where	where's
whereafter	whereas	whereby	wherein	whereupon	wherever
whether	which	while	whither	who	who's
whoever	whole	whom	whose	why	will
willing	wish	with	within	without	won't
wonder	would	would	wouldn't	x	y
yes	yet	you	you'd	you'll	you're
you've	your	yours	yourself	yourselves	z
zero					

Appendix C

Annotation for Japanese National Pension Act

C.1 Annotated Japanese Keyphrases

We list 208 Japanese keyprhases which is annotated in Japan National Pension Act as followings:

一時金	三乗根	世帯
世帯主	事務	事務所
事情	事故	事業
事由	事項	付加年金
住所	価額	保険料
保険料免除期間	保険料全額免除期間	保険料半額免除期間
保険料四分の一免除期間	保険料四分の三免除期間	保険料改定率
保険料率	保険料納付済期間	停止
傷病	全額	公的年金被保険者等総数
共済組合	共済組合等	加入員
加入者	労働基準法	効力
区長	医師	厚生労働大臣
厚生労働省令	厚生年金保険法	原因
収入	取得	受給権
受給権者	同種	名目手取り賃金変動率
名称	国家公務員共済組合法	国家公務員共済組合連合会
国民	国民年金事業	国民年金基金
国民年金基金連合会	国民年金手帳	地区内
地域型基金	地方公務員共済組合連合会	地方公務員等共済組合法

地方税法
基準傷病
基準年度以後改定率
基準障害
基金
場所
天災
夫
妻
委託
婚姻
婚姻関係
子
学生等
孫
寡婦年金
届出
市町村
市町村長
帳簿
年度
年金
年金保険者
年金給付
延滞金
徴収
徴収金
徴収金額
恩給法
所在
所在地
所得
手続
承認
掛金
措置
援助
損害賠償
支給
支給事由
改定
改定率
政令
政府
故意
施設
日本国内
日本私立学校振興・共済事業団
期間
期限
未支給
業務
標準報酬額等平均額
権利
歯科医師
死亡
死亡一時金
死亡日
母
氏名
法令
消滅
父
物価変動率
物価指数
状態
状況
理由
生死
生活保護法
生活扶助
生計
生計維持
申請
疾病
目的
督促
督促状
社会保険審査会
社会保険庁長官
福祉
私立学校教職員共済法
種別
積立金
第一号被保険者
第三号被保険者
第三者
第二号被保険者
管掌者
納付
納付事務
納付受託者
納付義務者
納期限
終了年度
組合員
組織
結果
給付
給付額
義務
老齢
老齢基礎年金
老齢給付
者
職務
職員
職能型基金
胎児
航行
行方不明

行為
被保険者
被保険者期間
被保険者等
被扶養配偶者
被用者年金保険者
被用者年金各法
被用者年金被保険者等
要件
規定
解除
認定
調整期間
調整率
請求
財政
財政の現況及び見通し
財政均衡期間
費用
資格
農業者年金
返還金債権
退職
運用職員
適用
遺族
遺族基礎年金
都道府県
配偶者
金銭
金額
限度
障害
障害基礎年金
障害状態
障害等級
障害認定日
離縁
額
養子

C.2 Annotated Sub-Ordinate Relations

The 226 annotated sub-ordinate relations in Japan National Pension Act are listed below:

法令 → null
生活保護法 → 法令
厚生年金保険法 → 被用者年金各法
地方公務員等共済組合法 → 被用者年金各法
労働基準法 → 法令
厚生労働省令 → 法令
農業者年金 → null
社会保険審査会 → 組織
世帯 → 組織
共済組合等 → 組織
地方公務員共済組合連合会 → 共済組合等
共済組合 → 共済組合等
被用者年金保険者 → 組織
年金保険者 → 組織
基金 → 国民年金基金
基金 → 組織
者 → null
恩給法 → 法令
被用者年金各法 → 法令
国家公務員共済組合法 → 被用者年金各法
私立学校教職員共済法 → 被用者年金各法
地方税法 → 法令
政令 → 法令
組織 → null
国民年金基金連合会 → 組織
政府 → 組織
日本私立学校振興・共済事業団 → 共済組合等
国家公務員共済組合連合会 → 共済組合等
管掌者 → 組織
納付受託者 → 組織
国民年金基金 → 基金
国民年金基金 → 組織
被扶養配偶者 → null
第三者 → null

医師 → null
職員 → null
国民 → null
胎児 → null
区長 → null
社会保険庁長官 → null
組合員 → null
加入者 → null
被保険者 → null
第二号被保険者 → 被保険者
被保険者等 → null
受給権者 → null
生計 → 状態
所得 → 状態
離縁 → 状態
婚姻 → 状態
死亡 → 生死
傷病 → 状態
疾病 → 傷病
障害 → 状態
状況 → null
未支給 → 状況
行為 → null
申請 → 行為
請求 → 行為
承認 → 行為
適用 → 行為
解除 → 行為
手続 → 行為
財政 → 行為
給付 → 行為
督促 → 行為
航行 → null
年金 → 年金給付
年金 → null
障害基礎年金 → 年金
遺族基礎年金 → 年金
歯科医師 → null
運用職員 → 職員
学生等 → null
市町村長 → null
厚生労働大臣 → null
世帯主 → null
加入員 → null
納付義務者 → null
第一号被保険者 → 被保険者
第三号被保険者 → 被保険者
被用者年金被保険者等 → 被保険者等
状態 → null
収入 → 状態
生計維持 → 状態
退職 → 状態
生死 → 状態
行方不明 → 状態
基準傷病 → 傷病
障害状態 → 状態
基準障害 → 障害
停止 → 状況
事情 → 状況
取得 → 行為
届出 → 行為
納付 → 行為
認定 → 行為
改定 → 行為
措置 → 行為
委託 → 行為
支給 → 行為
徴収 → 行為
消滅 → null
年金給付 → 年金
年金給付 → null
障害基礎年金 → 年金給付
遺族基礎年金 → 年金給付
寡婦年金 → 年金給付

寡婦年金 → 年金
死亡一時金 → 年金
老齢基礎年金 → 年金
付加年金 → 年金
老齢給付 → 年金
養子 → 子
父 → null
配偶者 → null
夫 → 配偶者
権利 → null
返還金債権 → 権利
公的年金被保険者等総数 → null
物価変動率 → null
基準年度以後改定率 → 改定率
保険料率 → null
三乗根 → null
金額 → 額
金額 → null
価額 → 金額
標準報酬額等平均額 → 金額
給付額 → 金額
徴収金額 → 金額
全額 → 金額
老齢 → null
財政均衡期間 → 期間
被保険者期間 → 期間
保険料全額免除期間 → 保険料免除期間
保険料四分の一免除期間 → 保険料免除期間
調整期間 → 期間
障害認定日 → null
死亡日 → null
納期限 → null
都道府県 → 場所
住所 → 場所
所在 → 場所
氏名 → 名称
国民年金手帳 → null
死亡一時金 → 年金給付
老齢基礎年金 → 年金給付
付加年金 → 年金給付
老齢給付 → 年金給付
子 → null
孫 → null
母 → null
妻 → 配偶者
遺族 → null
受給権 → 権利
年度 → null
名目手取り賃金変動率 → null
改定率 → null
保険料改定率 → 改定率
調整率 → null
物価指数 → null
額 → 金額
額 → null
価額 → 額
標準報酬額等平均額 → 額
給付額 → 額
徴収金額 → 額
全額 → 額
期間 → null
保険料納付済期間 → 期間
保険料免除期間 → 期間
保険料半額免除期間 → 保険料免除期間
保険料四分の三免除期間 → 保険料免除期間
限度 → null
期限 → null
終了年度 → null
場所 → null
市町村 → 場所
所在地 → 場所
名称 → null
帳簿 → null
督促状 → null

財政の現況及び見通し → null
金銭 → null
徴収金 → 金銭
延滞金 → 金銭
保険料 → 金銭
事業 → null
施設 → null
業務 → null
事務 → 業務
故意 → null
種別 → null
職能型基金 → null
損害賠償 → null
援助 → null
事故 → null
福祉 → null
地区内 → null
婚姻関係 → null
結果 → null
資格 → null
目的 → null
事由 → 理由
義務 → null
積立金 → 金銭
費用 → 金銭
掛金 → 金銭
一時金 → 金銭
国民年金事業 → 事業
事務所 → 施設
職務 → 業務
納付事務 → 事務
同種 → null
障害等級 → null
地域型基金 → null
生活扶助 → null
規定 → null
天災 → null
事項 → null
日本国内 → null
原因 → null
効力 → null
要件 → null
理由 → null
支給事由 → 事由

Publications

Journal Articles

[1] **Tho Thi Ngoc Le**, Kiyoaki Shirai, Minh Le Nguyen, Akira Shimazu. "Extracting Indices from Japanese Legal Documents." *Artificial Intelligence and Law*. (DOI: 10.1007/s10506-015-9168-8)

[2] **Tho Thi Ngoc Le**, Akira Shimazu, Minh Le Nguyen, Kiyoaki Shirai. "Unsupervised Keyphrase Extraction in General Domain: Introducing New Kinds of Words to Keyphrases." Submitted to *Information Processing & Management*.

Refereed International Conference Papers

[3] **Tho Thi Ngoc Le**, Minh Le Nguyen, Akira Shimazu. "Generalizing Hierarchical Structure of Indices for Japanese Legal Documents." In *Proceedings of the 19th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES)*, Procedia Computer Science, pages 103-–112, 2015.

[4] **Tho Thi Ngoc Le**, Minh Le Nguyen, Akira Shimazu. "Unsupervised Keyword Extraction for Japanese Legal Documents." In *Proceedings of Legal Knowledge and Information Systems - JURIX 2013: The Twenty-Sixth Annual Conference*, pages 97–106, 2013.

[5] **Tho Thi Ngoc Le**, Minh Le Nguyen, Akira Shimazu. "A Study on Hierarchical Table of Indexes for Multi-documents." In *Proceedings of the 8th International Conference on Natural Language Processing (JapTAL)*, LNCS/LNAI 7614, pages 222–227, 2012.

Refereed International Workshop Paper

[6] **Tho Thi Ngoc Le**, Minh Le Nguyen, Akira Shimazu. "A Hierarchy of Legal Indices." *The 8th International Workshop on Juris-informatics (JURISIN 2014).*